Algebraic Algorithms for Information Spreading

Thesis submitted in partial fulfillment
of the requirements for the degree of
"DOCTOR OF PHILOSOPHY"

by

Michael Borokhovich

Submitted to the Senate of Ben-Gurion University
of the Negev

September 30, 2013

Beer-Sheva

Algebraic Algorithms for Information Spreading

Thesis submitted in partial fulfillment
of the requirements for the degree of
"DOCTOR OF PHILOSOPHY"

by

Michael Borokhovich

Submitted to the Senate of Ben-Gurion University
of the Negev

Approved by the advisors:

Dr. Chen Avin: ____________ 30/04/2014

Prof. Zvi Lotker: ____________ 30/04/2014

Approved by the Dean of the Kreitman School of Advanced Graduate Studies:

Prof. Michal Shapira: ____________

April 30, 2014

Beer-Sheva

Research-Student's Affidavit when Submitting the Doctoral Thesis for Judgment

I Michael Borokhovich, whose signature appears below, hereby declare that (Please mark the appropriate statements):

V I have written this Thesis by myself, except for the help and guidance offered by my Thesis Advisors.

V The scientific materials included in this Thesis are products of my own research, culled from the period during which I was a research student.

V This Thesis incorporates research materials produced in cooperation with others, excluding the technical help commonly received during experimental work. Therefore, I am attaching another affidavit stating the contributions made by myself and the other participants in this research, which has been approved by them and submitted with their approval.

Date: 30/09/2013

Student's name: Michael Borokhovich

Signature:

The work was carried out under the supervision of
Dr. Chen Avin and Prof. Zvi Lotker.

In the Department of Communication Systems Engineering
Faculty of Engineering

Acknowledgments

I would like to express my sincere gratitude to my advisors Dr. Chen Avin and Prof. Zvi Lotker. They gave me a lot of professional knowledge, always encouraged me, and provided close guidance through the PhD process. Leading by example, they showed me how to gain fruitful collaborations and how to fearlessly pursue new research areas. But their mentoring was far beyond technical supervision. Chen and Zvi always were there to help me with my personal concerns regarding the research, public speaking, teaching, and career choices. I could not have wished for better advisors.

I gratefully acknowledge the generous financial support of my advisors and the Department of Communication Systems Engineering at Ben-Gurion University, Israel. Without it I would not have been able to accomplish this journey.

Many thanks to my collaborators Dr. Yoram Hadad, Dr. Erez Kantor, Ms. Merav Parter, and Prof. David Peleg. The part of the thesis about the power allocation in wireless networks would not have been possible without them.

I wish to thank my family. My late grandparents Abraham and Bronya, for the love they gave me, and whose academic achievements were always my inspiration. My parents Vladimir and Inna and my sister Svetlana for all their love, encouragement, and for believing in me in good and bad times. My mother-in-law, Marianna, for her continued help and support. And of course, my wonderful daughter Lily who gives me so much love and happiness every day. Finally, my special appreciation and thanks go to my lovely wife, Anna, who always helped and encouraged me during this challenging period, surrounding me with love and care.

This work is dedicated to my family

Vita

1979	Born, Saint Petersburg, Russia.
1990	Repatriated to Israel.
1997-2000	Military Service as Electronics Technician, IAF.
2001-2005	B.Sc., Communication Systems Engineering, Ben-Gurion University of The Negev, Beer Sheva, Israel.
2005-2007	Software Engineer, VocalTec Communications Inc.
2007-2009	M.Sc., Communication Systems Engineering, Ben-Gurion University of The Negev, Beer Sheva, Israel.
2009-present	Ph.D., Communication Systems Engineering, Ben-Gurion University of The Negev, Beer Sheva, Israel.

List of Publications

[1] Chen Avin, Michael Borokhovich, Keren Censor-Hillel, and Zvi Lotker. Order optimal information spreading using algebraic gossip. *Distributed Computing*, 26(2):99–117, 2013.

[2] Chen Avin, Michael Borokhovich, Yoram Haddad, Erez Kantor, Zvi Lotker, Merav Parter, and David Peleg. Generalized perron-frobenius theorem for multiple choice matrices, and applications. In *SODA*, pages 478–497, 2013.

[3] Chen Avin, Michael Borokhovich, Bernhard Haeupler, and Zvi Lotker. Self-adjusting grid networks to minimize expected path length. In *SIROCCO*, 2013.

[4] Chen Avin, Michael Borokhovich, Zvi Lotker, and David Peleg. Brief announcement: Distributed mst in core-periphery networks. In *DISC*, 2013.

[5] Chen Avin, Michael Borokhovich, and Stefan Schmid. Obst: A self-adjusting peer-to-peer overlay based on multiple bsts. In *P2P*, 2013.

[6] Michael Borokhovich and Stefan Schmid. How (not) to shoot in your foot with sdn local fast failover: A load-connectivity tradeoff. In *OPODIS*, 2013.

[7] Chen Avin, Michael Borokhovich, Yoram Haddad, and Zvi Lotker. Optimal virtual traffic light placement. In *FOMC*, pages 6:1–6:10, 2012.

[8] Michael Borokhovich, Chen Avin, and Zvi Lotker. Bounds for algebraic gossip on graphs. *Random Structures & Algorithms*, page doi:10.1002/rsa.20480, 2012.

[9] Chen Avin, Michael Borokhovich, Keren Censor-Hillel, and Zvi Lotker. Order optimal information spreading using algebraic gossip. In *PODC*, pages 363–372, 2011.

[10] Chen Avin, Michael Borokhovich, Asaf Cohen, and Zvi Lotker. Efficient distributed source coding for multiple receivers via matrix sparsification. In *ISIT*, pages 2045–2049, 2011.

[11] M. Borokhovich, C. Avin, and Z. Lotker. Tight bounds for algebraic gossip on graphs. In *ISIT*, pages 1758 –1762, 2010.

[12] Chen Avin, Michael Borokhovich, and Arik Goldfeld. Mastering (virtual) networks - a case study of virtualizing internet lab. In *CSEDU (2)*, pages 250–257, 2009.

Contents

List of Figures

Abstract of the Thesis

Algebraic Algorithms for Information Spreading

By: **Michael Borokhovich**

Advisors: **Dr. Chen Avin, Prof. Zvi Lotker**

April 30, 2014

We study problems of information spreading in communication networks using algebraic methods and algorithms. First, we study the stopping times of *uniform* gossip algorithms for network coding, where *uniform* means that a communication partner is always chosen uniformly at random. We analyze algebraic gossip (i.e., random linear network coding) and consider three gossip algorithms for all-to-all information spreading: `PULL`, `PUSH`, and `EXCHANGE`. The stopping time of algebraic gossip for all-to-all dissemination is known to be linear for the complete graph, but the question of determining a tight upper bound or lower bounds for general graphs remained open prior to our work. We take a major step in solving this question, and prove that algebraic gossip on any graph of size n is $O(\Delta n)$ where Δ is the maximum degree of the graph. This leads to a tight bound of $\Theta(n)$ for bounded degree graphs and an upper bound of $O(n^2)$ for general graphs. We show that the latter bound is tight by providing an example of a graph with a stopping time of $\Omega(n^2)$. Our proofs use a novel method that relies on Jackson's queuing theorem to analyze the stopping time of network coding; this technique is likely to become useful for future research.

Then, we extend our study of gossip-based information spreading. We use algebraic gossip to disseminate k distinct messages to all n nodes in a network (k-dissemination problem). For arbitrary networks we provide a new upper bound for

uniform algebraic gossip of $O((k + \log n + D)\Delta)$ rounds with high probability, where D and Δ are the diameter and the maximum degree in the network, respectively. For many topologies and selections of k this bound improves previous results, in particular, for graphs with a constant maximum degree it implies that uniform gossip is *order optimal* and the stopping time is $\Theta(k + D)$. To eliminate the factor of Δ from the upper bound we propose a non-uniform gossip protocol, TAG, which is based on algebraic gossip and an arbitrary spanning tree protocol $\mathcal{S}$. The stopping time of TAG is $O(k + \log n + d(\mathcal{S}) + t(\mathcal{S}))$, where $t(\mathcal{S})$ is the stopping time of the spanning tree protocol, and $d(\mathcal{S})$ is the diameter of the spanning tree. We provide a general case in which this bound leads to an order optimal protocol. For $k = \Omega(n)$, using a simple gossip broadcast protocol that creates a spanning tree in at most linear time, we show that TAG finishes after $\Theta(n)$ rounds for any graph.

Finally, we turn to another interesting problem. Consider n receivers and m ($m \geq n$) transmitters embedded in $\mathbb{R}^d$. Each receiver has a certain number of dedicated transmitters that can transmit to it (and only to it) synchronously. All transmitters are set to transmit at the same time with the same frequency, thus causing interference to the other receivers. Therefore, receiving and decoding a message at receiver r_i depends on the transmitting power of its dedicated transmitters (the desired signal) as well as the power of the rest of the transmitters (the interference). We assume the SINR model for communication channel, i.e., if the signal strength received by a device divided by the interfering strength of other simultaneous transmissions is above some *reception threshold* β, then the receiver successfully receives the message, otherwise it does not. The question of power control is then to find an optimal power assignment for the transmitters, so as to make β as high as possible and ease the decoding process. We show that there exists an optimal power allocation in which for every receiver only one dedicated transmitter transmits (so-called $\mathbf{0^*}$ solution), and give a polynomial time algorithm that uses nontrivial algebraic tools for finding it.

Chapter 1

Introduction

1.1 Preface

The thesis presents main results of the work done during my Ph.D. studies (2009-2013). We study the problems of information spreading in networks using algebraic and algorithmic approaches. The research deals with different aspects and models. We consider uniform and non-uniform gossip algorithms, synchronous and asynchronous time models, and various message exchange variations (`PULL`, `PUSH`, `EXCHANGE`). Then we also consider a wireless network with the *signal-to-interference* connectivity model and solve the problem of optimal power allocation. All presented results use original techniques that probably would be useful in future research.

The lion's share of the thesis is dedicated to the analysis of gossip dissemination algorithms and, in particular, algebraic gossip, which is a combination of gossip and random linear network coding [36]. The work on algebraic gossip resulted in two journal publications: the first in "Random Structures & Algorithms" (RSA) [9] and the second in "Distributed Computing" (DIST) [3]. Preliminary versions of these papers appeared in the Proceedings of ISIT 2010 [8] and PODC 2011 [2] conferences, respectively.

During the course of my Ph.D. studies, there were fruitful collaborations that led to additional publications. A very interesting work, which was done in collaboration with Dr. Yoram Hadad, Dr. Erez Kantor, Ms. Merav Parter, and Prof. David Peleg, was devoted to generalizing the famous Perron-Frobenius Theorem to non-square matrices [4]. In this thesis, we show an application of the Generalized Perron-Frobenius Theorem, proved in [4], for the power allocation problem in wireless networks, where each receiver has several dedicated transmitters. Additional interesting research results that are not described in the thesis, can be found in the full list of my publications.

1.2 Gossip Algorithms

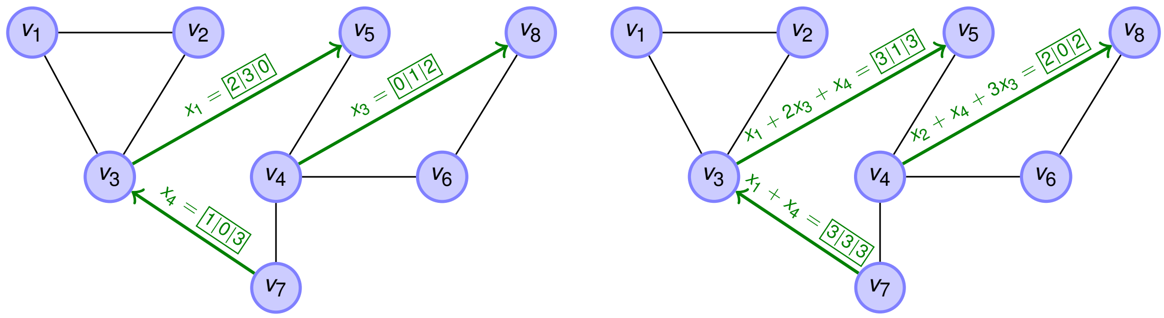
Ad hoc and sensor networks usually do not have a central entity for managing information spreading. Moreover, such wireless stations have limited energy and computational power. All this leads to a need for local, distributed, and efficient algorithms for disseminating information across the network. Gossip algorithms for information dissemination were first introduced in [24] in the context of updating a database replicated at many sites. In [23], Amazon.com introduced Dynamo, a highly available and scalable data store, used for storing the state of a number of core services. Dynamo employs a gossip-based distributed failure detection and membership protocol. Gossip approach was also used by Harchol-Balter et al. [35] for developing an efficient and distributed resource discovery algorithm. This algorithm was later adopted by Akamai Technologies for machines discovery in content distribution systems. In a gossip algorithm there is no need for a centralized controller and every node relies only on its local information; thus gossip algorithms are inherently distributed. A systematic survey of gossip algorithms used in communication networks can be found in [54].

Let us now briefly describe some specific properties and assumptions that we used regarding gossip algorithms. First, we define two time models: synchronous and asynchronous. In a synchronous time model, in each *round*, each node chooses a single neighbor as the communication partner and takes an action. In an asynchronous time model, at every *timeslot*, a single node wakes up, chooses a communication partner, and takes an action. We say that every n consecutive *timeslots* are considered as one *round*. The action taken by the nodes can be one of the following variations: `PUSH`, `PULL`, or `EXCHANGE`; where in `PUSH`, one message is sent to the partner, in `PULL`, one message is received from the partner, and in `EXCHANGE` (sometimes called `PUSH-PULL` in the literature), one message is sent and one is received. We assume that all messages have limited size (i.e., a node may not be able to send all its data in one message). If the communication partner is chosen uniformly at random, we obtain a uniform gossip. In this thesis we consider both uniform and non-uniform gossip, both time models (synchronous and asynchronous), and all the gossip variations (`PUSH`, `PULL`, and `EXCHANGE`).

1.3 Algebraic Gossip Protocol

We distinguish gossip *algorithms* and gossip *protocols* [10]. While a gossip algorithm defines when and to which neighbor a message will be sent, a gossip protocol defines the task that should be performed by the network using a gossip algorithm, e.g., calculation of some aggregate function, or information dissemination. In particular, a gossip protocol defines the content of each message and the stopping condition for each node.

Algebraic gossip is a gossip protocol that can perform information dissemination tasks. For example, assume that every node in the network has one initial message, and the task is to deliver all these messages to all the nodes (this scenario is called *all-to-all* dissemination). In a simple gossip dissemination protocol, a node would forward one (since the message size is limited) uncoded message from the set of messages it has collected so far to a neighbor according to the gossip algorithm. In the algebraic gossip protocol, instead of sending uncoded messages, a node builds a random linear combination (equation) of the messages (which are, in turn, also linear combinations



(a) (b)

Figure 1.1: In this example, every initial message is a vector (of length 3) of elements from the finite field $\mathbb{F}_4$. (a) – Simple gossip dissemination protocol. A node sends uncoded messages $x_i \in \mathbb{F}_4^3$. (b) – Algebraic gossip. A node sends random linear combinations $\sum_i a_i x_i$ of the messages it has collected, along with the random coefficients $a_i \in \mathbb{F}_4$. Note that $\sum_i a_i x_i$ is a vector in $\mathbb{F}_4^3$.

of the initial messages) it has collected so far, and sends them according to the gossip algorithm (see Figure 1.1 for an illustration). Once a node accumulates enough independent equations (i.e., when the number of independent equations is equal to the number of initial messages that needed to be spread), it can decode all the initial messages by simply solving the linear system. When sending a random linear combination, a node has to send both the outcome of the combination and all the random coefficients. All the algebraic operations are performed over a finite field and thus the size of the combination's outcome is the same as the size of a single message. Moreover, if the message size is relatively large, the overhead of sending the coefficients becomes negligible.

Uniform algebraic gossip was first proposed by Deb *et al.* [22]. The authors considered synchronous time model, `PUSH` and `PULL` gossip variations, and studied it on the complete graph. They showed that algebraic gossip strictly outperforms an RMS (Random Message Selection) gossip protocol in which a node sends a randomly selected message from the messages it has collected so far. The key reason for the advantage of algebraic gossip is that by using random linear network coding (i.e., sending random linear combinations of messages it has), a node is able to send a *helpful* message to a neighbor, with higher probability than in RMS.

So, Deb *et al.* [22] analyzed the stopping time of algebraic gossip for a complete graph. Mosk-Aoyama and Shah [47] analyzed uniform algebraic gossip for arbitrary graphs, but their bounds were not tight. The question of a tight upper bound remained open until our work, presented in Chapter 2.

We analyze algebraic gossip for an *all-to-all* dissemination task, and then we extend our results to the *many-to-all* case. While uniform algebraic gossip turns out to be an efficient dissemination protocol, there are graphs for which it performs badly. So, we go beyond the uniform gossip by proposing a nonuniform algebraic gossip protocol and prove that it is an order optimal dissemination protocol for many graph families.

1.4 Power Allocation Under SIR Model

Motivated by algorithmic aspects of wireless networks in the SINR model [6, 38, 5], we started to investigate the problem of power allocation when each receiver has several dedicated transmitters. In a classic SISO (single input single output) case, each receiver has exactly one dedicated transmitter. Here, we consider the MISO (multiple input single output) case, in which for every receiver we can transmit from several transmitters (see Figure 1.2 for an illustration). We assume that transmitters dedicated to the same receiver are perfectly synchronized and transmit exactly the same information; thus their signals just sum up at the receiver.

In the SINR model for communication channel, if the signal strength received by a device divided by the interfering strength of other simultaneous transmissions plus the white noise, is above some *reception threshold* β, then the receiver successfully

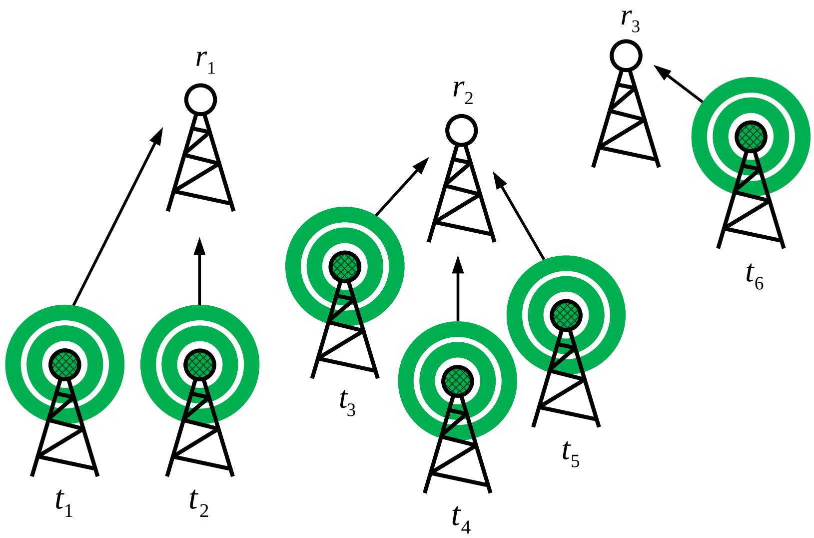


Figure 1.2: Example of a MISO wireless network. Receiver r_1 has two dedicated transmitters $\{t_1, t_2\}$, r_2 has three dedicated transmitters $\{t_3, t_4, t_5\}$, and receiver r_3 has a single dedicated transmitter t_6.

receives the message, otherwise it does not [52]. So, the goal of power control is finding the optimal power levels for transmitters in order to make β as high as possible, which will ease the decoding process and make it cheaper. In this work, we assume that there is no white noise, so, we will usually call our SINR model, the SIR (signal-to-interference) model.

In the SISO case, the power control problem was solved elegantly by Zander [56], who proved, using the well-known Perron–Frobenius Theorem [51, 30], that the optimal SIR ratio and the optimal power vector are closely related to the largest eigenvalue and the corresponding Eigen-vector of the *square* matrix representing the gains between each receiver-transmitter pair. But in our MISO case, the matrix of gains is no longer *square* (since each receiver may have more than one dedicated transmitter), and thus the Perron–Frobenius Theorem is not applicable. This motivated us to prove, in [4], the Generalized Perron–Frobenius Theorem. In this thesis, we use this generalized result to solve the power allocation problem for the MIMO case.

1.5 Main Research Questions and Results of the Thesis

Our first research question is an upper bound on the stopping time of algebraic gossip for the *all-to-all* dissemination task on an arbitrary, connected graph G_n with n nodes. For the *all-to-all* dissemination task, we assume that there are n initial messages, and each node possesses exactly one initial message. All the n messages are need to be delivered to all n nodes. Formally, we ask the following question:

Question 1. *What is the number of communication rounds it takes to complete the* ***all-to-all*** *dissemination task using algebraic gossip, in both synchronous and asynchronous time models, and for three gossip variations:* `PUSH`*,* `PULL`*, and* `EXCHANGE`*?*

Let Δ be the maximum degree of the graph G_n. We prove an upper bound for any network topology that is $O(\Delta n)$. This leads to a tight bound of $\Theta(n)$ for bounded degree graphs and an upper bound of $O(n^2)$ rounds for general graphs. Our proofs use a novel technique that relies on queuing theory to analyze the stopping time of algebraic gossip, and it is likely to become useful for future research.

The next question that we novelly raise in the context of algebraic gossip, is about the worst case lower bound, i.e., what is the worst topology for the stopping time of uniform algebraic gossip?

Question 2. *For the* worst case *graph, what is the* ***lower bound*** *on the number of communication rounds it takes to complete the* ***all-to-all*** *dissemination task using algebraic gossip, in both synchronous and asynchronous time models? What is the* worst case *graph?*

It turns out that the worst case stopping time is $\Omega(n^2)$, which shows that our upper bound of $O(n^2)$ for general graphs is tight. An example of such a worst case topology is the barbell graph (see Figure 1.3), due to its specific bottleneck that interconnects the two cliques.

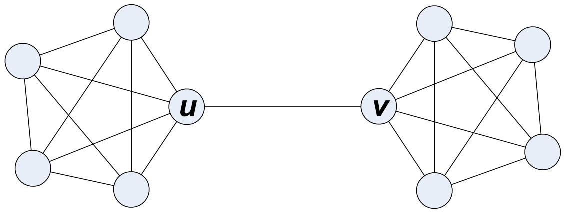


Figure 1.3: Barbell graph: two cliques of size $n/2$ connected with a single edge.

An additional contribution of this part of the thesis is the analysis of the `EXCHANGE` gossip variation. While traditionally algebraic gossip protocols used `PUSH` or `PULL` as their gossip algorithms, it was unclear if using `EXCHANGE` (that uses twice as many messages than `PULL` or `PUSH`) can lead to significant improvements in stopping time.

Question 3. *Can `EXCHANGE` gossip variation provide significant (in an order of magnitude) speedup for algebraic gossip, in comparison to `PUSH` and `PULL`?*

We give a positive answer to this question and prove that, for some topologies, using the `EXCHANGE` algorithm can be unboundedly better than using `PULL` or `PUSH`. We show that while the time it takes the `EXCHANGE` algorithm to complete the algebraic gossip on the star graph (which is a tree of n nodes with one node having degree $n-1$ and the other $n-1$ nodes having degree 1) is $O(n)$, the time it takes the `PULL` and `PUSH` algorithms to finish the same task, is $\Omega(n \log n)$. On the contrary, there are many other graphs (e.g., bounded degree graphs, complete graph) on which these three gossip variations have the same asymptotical behavior.

Then, we extend our study on algebraic gossip and consider a more general scenario of *many-to-all* dissemination. In this task, k initial messages located somewhere in the network (i.e., each node can possess an arbitrary number of initial messages), needed to be disseminated to all the nodes. As in the *all-to-all* case we assume an arbitrary, connected graph G_n with n nodes and maximum degree Δ.

Question 4. *What is the number of communication rounds it takes to complete the* ***many-to-all*** *dissemination task using algebraic gossip, in both synchronous and asynchronous time models?*

Let D represent the diameter of the graph G_n. We prove that an upper bound for delivering k messages to all the n nodes is $O((k + \log n + D)\Delta)$ rounds. For many topologies and selections of k this bound improves previous results, in particular, for graphs with a constant maximum degree it implies that uniform gossip is *order optimal* and the stopping time is $\Theta(k + D)$.

While algebraic gossip protocol achieves order optimal time for information dissemination for many topologies, there are graphs for which algebraic gossip performs badly (e.g., for the barbell graph in Figure 1.3). Hence, we were interested in finding an enhancement for algebraic gossip that will allow it to become optimal for many more cases.

Question 5. *Can we modify the algebraic gossip protocol (possibly by using a* ***non-uniform*** *gossip) so that it will become optimal for the information dissemination task?*

The answer to this question is a very important result in the context of gossip protocols. We propose a modified algebraic gossip protocol – TAG. This protocol is based on algebraic gossip and an arbitrary spanning tree protocol $\mathcal{S}$. First, using $\mathcal{S}$, TAG constructs a spanning tree on G_n, and then, non-uniform algebraic gossip is performed over this spanning tree. Let $t(\mathcal{S})$ be the stopping time of the spanning tree protocol $\mathcal{S}$, and $d(\mathcal{S})$ be the diameter of the resulting spanning tree. Then, the number of communication rounds needed to deliver k initial messages to all the n nodes is $O(k + \log n + d(\mathcal{S}) + t(\mathcal{S}))$. Additionally, we prove that using a simple gossip broadcast protocol that creates a spanning tree in at most linear time, TAG finishes after $\Theta(n)$ rounds for any graph that is *order optimal* for the case where $k = \Omega(n)$. Since TAG is based on a tree topology, which is vulnerable to edge failures, there is a question of how suitable TAG is for real-life networks. We discuss this issue in Section 3.4.3 and also propose there some natural enhancements that make TAG a more robust protocol.

While our bounds on the **uniform** algebraic gossip use the parameter Δ (maximum degree of the graph), which is very convenient to use and easy to find, it does not perfectly capture the stopping time behavior for every graph. While our bounds are tight in the worst case sense, there are topologies for which the bounds with Δ are not tight (e.g., the complete graph). Hence, in our first work on algebraic gossip [8] we raised a question of finding an alternative (to the maximum degree) graph feature that will better describe the stopping time for specific topologies. A recent work of Haeupler [33] successfully answers this question by using a min-cut measure of a graph, and a novel and elegant approach to prove the results. The results of [33] are tight for the *all-to-all* dissemination. However, for *many-to-all* dissemination, the results are not tight for specific graph families, while our bound of $O((k + \log n + D)\Delta)$ is tight. Further discussion and comparison can be found in Section 3.1.2.

Our last result presented here, deals with finding an optimal power allocation for a set of transmitters. Consider a wireless network of n receivers and m ($m \geq n$) transmitters. Each receiver has a certain number of dedicated transmitters that can transmit to it (and only to it) synchronously. All transmitters are set to transmit at the same time with the same frequency, thus causing interference to the other receivers. Assuming that there is no white noise, we define an SIR (*signal-to-interference*) ratio for a receiver as a ratio between the sum of the signals of dedicated transmitters divided by the sum of signals of all the other transmitters in the network. Let $g(i, j)$ be the gain of a signal transmitted by transmitter t_j at the location of the receiver r_i. In the SIR model, the energy of a signal fades with the distance to the power of the path-loss parameter α (which usually equals 2), i.e., $g(i, j) = \mathrm{d}(r_i, t_j)^{-\alpha}$. Let T denote the set of all the m transmitters in the network, and T_i denote the set of transmitters dedicated to the receiver r_i (for example, in Figure 1.2, $T_2 = \{t_3, t_4, t_5\}$). We denote by vector $\overline{X} = (X_1, X_2, \ldots, X_m)$ the power levels of all the m transmitters. Now we can formulate our research question.

Question 6. *Given a wireless network where each receiver has several dedicated transmitters, what is the optimal power allocation that achieves the highest possible* ***signal-to-interference*** *ratio?*

Formally, we have the following optimization problem:

$$\text{maximize } \beta \text{ subject to:} \tag{1.1}$$
$$\frac{\sum_{t_j \in T_i} X_j \cdot \mathrm{d}(r_i, t_j)^{-\alpha}}{\sum_{t_j \in T \setminus T_i} X_j \cdot \mathrm{d}(r_i, t_j)^{-\alpha}} \geq \beta \quad \forall i \in [1, \ldots, n],$$
$$\left\|\overline{X}\right\|_1 = 1, \quad \overline{X} \geq \overline{0}.$$

So, our goal is to find an optimal power allocation vector $\overline{X}^*$, such that the minimal *signal to interference* level (among all the receivers) in the network will be maximal. This optimization problem is not convex and not even log-convex [4], hence, there is a need for more creative methods for solving it. We first use the Generalized Perron-Frobenius Theorem, proved in [4], to characterize the optimal solution. Interestingly, it turns out that there exists an optimal power allocation in which only one transmitter per receiver can be active (so called **0*** solution), i.e., the option to have many dedicated transmitters, instead of one, does not improve the signal-to-interference ratio. The benefit one can get from having multiple transmitters is translated to the *choice* of the "best" set of single transmitters among exponentially many options. We then propose a nontrivial algorithm, which uses sophisticated algebraic tools, for finding the optimal solution. For a wireless network with n receivers, m transmitters, and maximum value of gain $\mathcal{G}_{max} = \max_{i,j} \{g(i,j)\}$, the running time of our algorithm is $O(n^3 \cdot T_{ellips} \cdot (\log(n \cdot \mathcal{G}_{max}) + n))$, where T_{ellips} is the running time of the Ellipsoid method [42] for checking a given β for feasibility. Notice that for a given

β, the optimization problem 1.1 becomes linear and thus T_{ellips} is also linear. Hence, we obtain a polynomial time algorithm for a non-convex (and even non-log-convex) optimization problem 1.1.

1.6 Organization

The thesis is organized as follows. Chapter 2 presents the analysis of algebraic gossip for *all-to-all* information dissemination. We provide tight upper and lower bounds on the running time of the algorithm. We also novelly show that using the `EXCHANGE` gossip variation, instead of the traditionally used `PUSH` or `PULL`, can be unboundedly better. Chapter 3 continues the study of gossip algorithms and presents the analysis of algebraic gossip for *many-to-all* dissemination and also proposes the alternative dissemination protocol that is order optimal for some general cases. In Chapter 4 we analyze the power allocation problem in which the information should be spread from a set of cooperating transmitters to the appropriate receiver in the presence of interference. We give an algebraic characterization of optimal solution and, using its algebraic properties, propose a polynomial time algorithm to find it.

The chapters are written in a way that allows independent reading, but for better understanding of algebraic gossip it is desirable to start with Chapter 2. Each chapter includes its own introduction, overview of the related literature, preliminaries, main results with all the proofs, and a conclusion section.

Chapter 2

Algebraic Gossip – All-to-All Dissemination

2.1 Introduction

Randomized gossip-based protocols are attractive due to their locality, simplicity, and structure-free nature, and have been offered in the literature for various tasks, such as ensuring database consistency and computing aggregate information [39, 40, 10]. Consider the case of a connected network with n nodes, each holding a value it would like to share with the rest of the network. Motivated by wireless networks and limited resource sensor motes, in recent years researchers have studied the use of randomized gossip algorithms together with network coding for this multicast task [45, 43]. The use of network coding protocols for multicast has received growing attention due to the ability of such protocols to significantly increase network capacity.

Let us look at the following basic network coding example [29]. Consider a *Butterfly Network* (see Figure 2.1) with sources S_1 and S_2, each is wishing to multicast to both R_1 and R_2. All links have capacity 1. Without network coding, the maximum achievable source-destination rate (assuming both rates are equal) is 1.5, due to a bottleneck at node C. Using a simple linear network coding, we can "xor" the information coming from S_1 and S_2 at the node C. By doing so, each receiver will

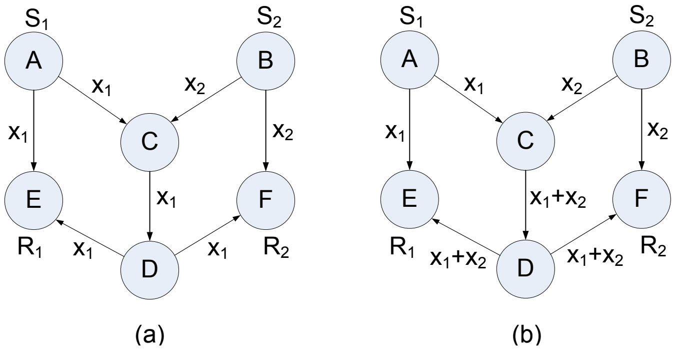


Figure 2.1: (a) – Without network coding, bottleneck at C brings the capacity to 1.5. (b) – With network coding, C transmits at each time unit information that is *helpful* to both receivers, thus the capacity is 2.

obtain two linear equations: R_1 will get: x_1 and $x_1 + x_2$, and R_2 will get: x_2 and $x_1 + x_2$. Now, each receiver is able to solve this simple linear system and discover x_1 and x_2. It is clear that the resulting throughput is now 2.

In this work we consider *algebraic gossip*, a gossip-based network coding protocol known as random linear coding [36]. In the discussion on gossip-based protocols we distinguish between the *gossip algorithm* and the *gossip protocol*. A gossip algorithm is a communication scheme in which at every timeslot, a random node chooses a random neighbor to communicate with. We consider three known gossip algorithms: `PUSH`: a message is sent *to* the neighbor, `PULL`: a message is sent *from* the chosen neighbor, and `EXCHANGE`: the two nodes exchange messages. The gossip protocol, on the other hand, determines the *content* of messages sent. In algebraic gossip protocol, the content of the messages is a random linear combination of all messages stored by the sending node. Once a node has received enough independent messages (independent linear equations) it can solve the system of linear equations and discover all the initial values of all other nodes.

We study the performance of algebraic gossip on arbitrary network topologies, where information is disseminated from all nodes in the network to all nodes, i.e., all-to-all dissemination. Previously, algebraic gossip was considered with `PUSH` and `PULL` gossip algorithms; here we also study the use of `EXCHANGE`, which can lead to significant improvements for certain topologies (as we show). Our main goal is to find tight bounds for the stopping time of the algebraic gossip protocol, both in expectation and with high probability (w.h.p.), i.e., with probability of at least $1 - \frac{1}{n}$.

2.1.1 Related Work

The stopping time question, i.e., bounding the number of rounds until protocol completeness, has been addressed in the past. Deb *et al.* [22] studied algebraic gossip using `PULL` and `PUSH` on the complete graph and showed a tight bound of $\Theta(n)$, a linear stopping time, both in expectation and with high probability. Boyd *et al.* [10, 12] studied the stopping time of a gossip protocol for the *averaging problem* using the `EXCHANGE` algorithm. They gave a bound for symmetric networks that is based on the second largest eigenvalue of the transition matrix or, equally, the mixing time of a random walk on the network, and showed that the mixing time captures the behavior of the protocol. Mosk-Aoyama and Shah [47] used a similar approach to [10] and [12] to analyze algebraic gossip on arbitrary networks. They consider symmetric stochastic matrices that (may) lead to a non-uniform gossip and gave an upper bound for the `PULL` algorithm that is based on a measure of conductance of the network. As the authors mentioned, the offered bound is not tight, which indicates that the conductance-based measure does not capture the behavior of the protocol. The question about a worst case topology for algebraic gossip was not previously addressed in the literature.

2.1.2 Overview of Results of the Current Chapter

The main contribution of this chapter is new bounds for the stopping time of algebraic gossip for all-to-all dissemination on arbitrary graphs. Our bounds are tight for many graph families; moreover, for almost any chosen maximum degree there exist graphs for which the bounds are tight. First, in Theorem 2.4, we show that there exists a family of graphs for which algebraic gossip takes $\Omega(n^2)$ rounds. Our main result then, Theorem 2.1, gives an upper bound of $O(\Delta n)$ for the stopping time of algebraic gossip on any graph, where Δ is the maximum degree in the graph.

Theorem 2.1. *For the asynchronous (synchronous) time model and for any graph G of size n with maximum degree Δ, the stopping time of algebraic gossip is $O(\Delta n)$ rounds both in expectation and with high probability.*

This result immediately leads to two interesting corollaries. In Corollary 2.1 we state a matching upper bound: for any graph of size n, since the max $\Delta = n$, algebraic gossip will stop w.h.p. in $O(n^2)$ rounds. In Corollary 2.2 we conclude a strong tight bound for *any* constant degree network (i.e., Δ is constant) of $\Theta(n)$. This improves upon known previous upper bounds that, for certain constant degree graphs, had an upper bound of $O(n^2)$. Note that the bound of $O(\Delta n)$ is not tight for all graphs (e.g., the complete graph) and the question of determining the properties of a network that capture tightly the stopping time of algebraic gossip is still open. We also show in Theorem 2.4 that the upper bound $O(\Delta n)$ is tight in the sense that for almost any Δ there exist graphs for which algebraic gossip takes $\Omega(\Delta n)$ rounds.

The second contribution of the chapter is the technique we use to prove our results. We novelly bound the stopping time of algebraic gossip via reduction to a network of queues and by the *stationary* state of the network that follows form *Jackson's theorem* for an open network of queues. The idea of using a queuing theory approach for network coding analysis was first introduced in [44] but, as opposed to our approach, it did not include the aspect of a gossip communication model. We believe that the type of reduction presented in this work could be used for future analysis of gossip protocols.

Third, we compare three gossip algorithms: `PUSH`, `PULL`, and `EXCHANGE`. While traditionally algebraic gossip used PULL or PUSH as its gossip algorithms, it was unclear if using EXCHANGE (that uses twice as many messages than PULL or PUSH) can lead to significant improvements in stopping time. We give a surprising affirmative answer to this question and prove that, for some topologies, using the EXCHANGE algorithm can be unboundedly better than using PULL or PUSH. We show that while the time it takes the EXCHANGE algorithm to complete the algebraic gossip on the star graph is linear, i.e., $O(n)$ the time it takes the PULL and PUSH algorithms to finish the same task, is $\Omega(n \log n)$. On the contrary, there are many other graphs such as the complete graph and all constant maximum degree graphs (see Section 2.4), on which these three gossip algorithms have the same asymptotical behavior.

Recently, there have been some advances in answering open questions raised in this work. In particular, the conference paper of Haeupler [33] and our recent work presented in Chapter 3. We discuss these works in Conclusions.

The rest of the chapter is organized as follows. In Section 2.2 we present the communication and time models, define gossip algorithms and gossip protocols, and formally state the *gossip stopping problem*. In Section 2.3 we show that algebraic gossip on the ring graph is linear using Jackson's theorem. In Section 2.4 we prove our main results: a tight upper bound for arbitrary networks and a tight linear bound for graphs with a constant maximum degree. Section 2.5 gives an answer to the question: "Can EXCHANGE be better than PUSH or PULL?" by providing a topology for which EXCHANGE is unboundedly faster. We conclude in Section 2.8.

2.2 Preliminaries and Models

2.2.1 Network and Time Model

We model the communication network by a connected undirected graph $G_n = G_n(V, E)$, where $V = \{v_1, v_2, ..., v_n\}$ is the set of vertices and $E \subseteq V \times V$ is the set of edges. Let

$N(v) \subseteq V$ be a set of neighbors of node v and $d_v = |N(v)|$ its degree, let $\Delta = \max_v d_v$ be the maximum degree of G_n.

The time is assumed to be slotted where n consecutive timeslots are regarded as one *round*. We consider the following time models:

- **Asynchronous time model.** At every timeslot, a node selected independently and uniformly at random takes an action (determined by a Gossip Algorithm) and a single pair of nodes communicates.[1] In this model there is no guarantee that a node will be selected exactly once in a round; nodes can be selected several times or not at all.

- **Synchronous time model.** At every round, all the nodes wake up synchronously and every node takes an action (determined by a Gossip Algorithm).

2.2.2 Gossip Algorithms

A *gossip algorithm* defines the way information is exchanged or spread in the network. When a node wakes up (according to a time model), it takes an information spreading action that is divided into two phases: (i) choosing a communication partner and (ii) spreading the information. A *communication partner* $u \in N(v)$ is chosen by node $v \in V$ with probability p_{vu}. Throughout this chapter we will assume *uniform gossip algorithms*, i.e., $p_{vu} = \frac{1}{d_v}$.

We distinguish three gossip algorithms for information spreading between v and u, `PUSH`, `PULL`, and `EXCHANGE` as explained in the Introduction. We assume that in the asynchronous time model, messages sent in timeslot t are received in timeslot t and can be forwarded or processed at timeslot $t + 1$, and in the synchronous time model, messages sent in round t are received in round t and can be forwarded or processed at round $t + 1$.

[1] Alternately, this model can be seen as each node having a clock that ticks at the times of a rate 1 Poisson process and there is a total of n clock ticks per round [10].

2.2.3 Algebraic Gossip

A *gossip protocol* is a task that is being executed using gossip algorithms, for example, calculation of aggregate functions, resource discovery, and database consistency. We now describe the algebraic gossip protocol for the multicast task: disseminating n initial values of the nodes to all n nodes.

Let $\mathbb{F}_q$ be a field of size q, each node $v_i \in V$ holds an initial value x_i that is represented as a vector in $\mathbb{F}_q^r$. We can represent every message as an integer value bounded by M, and therefore, $r = \lceil \log_q(M) \rceil$. All transmitted messages have a fixed length and represent linear equations over $\mathbb{F}_q$. The variables of these equations are the initial values $x_i \in \mathbb{F}_q^r$ and a message contains the coefficients of the variables and the result of the equation; therefore the length of each message is: $r \log_2 q + n \log_2 q$ bits. A message is built as a random linear combination of all messages stored by the node and the coefficients are drawn uniformly at random from $\mathbb{F}_q$. A received message will be appended to the node's stored messages only if it is independent of all linear equations (messages) that are already stored by the node and otherwise it is ignored. Initially, node v_i has only one linear equation that consists of only one variable corresponding to x_i multiplied by a coefficient 1 and equal to the value of x_i, i.e., the node knows only its initial value. Once a node receives n independent equations it is able to decode all the initial values and thus completes the task.

For a node v at timeslot (round)[2] t, let $S_v(t)$ be the subspace spanned by the linear equations (or vectors) it stores (i.e., the coordinates of each vector are the coefficients of the equation) at the beginning of timeslot (round) t. The dimension (or rank) of a node is the dimension of its subspace, i.e., $dim(S_v(t))$ and it is equal to the number of independent linear equations stored by the node.

We say that a node v is a ***helpful node*** to node u at the timeslot (round) t if and only if $S_v(t) \not\subset S_u(t)$, i.e., iff a random linear combination constructed by v can

[2]For asynchronous time model – timeslot, for synchronous – round.

be linearly independent with all equations (messages) stored by u. We call a message a ***helpful message*** if it increases the dimension of the node. The following lemma, which is a part of Lemma 2.1 in [22], gives a lower bound for the probability of a message sent by a *helpful node* to be a *helpful message*.

Lemma 2.1 ([22])**.** *Suppose that node v is* helpful *to node u at the beginning of the timeslot (round) t. If v transmits a message to u at the timeslot (round) t, then:*

$$\Pr\left(dim(S_u(t+1)) > dim(S_u(t))\right) \geq 1 - \tfrac{1}{q}.$$

That is, the probability of the message to be helpful is at least $1 - \frac{1}{q}$.

2.2.4 The Gossip Stopping Problem

Our goal is to compute bounds on time and number of messages needed to be sent in the network to complete various gossip protocols over various gossip algorithms. For this purpose we define the following:

Definition 2.1 (Excepted and high probability stopping times)**.** *Given a graph G_n, gossip algorithm $\mathcal{A}$, and a gossip protocol $\mathcal{P}$, the stopping time $t(\mathcal{A}, \mathcal{P}, G_n)$ is a random variable defined as the number of timeslots by which all nodes complete the task. $E[t(\mathcal{A}, \mathcal{P}, G_n)]$ is the expected stopping time and the* high probability stopping time $\hat{t}$ *is defined as follows:*

$$\hat{t}(\mathcal{A}, \mathcal{P}, G_n) = \min_{t \in \mathbb{Z}} \left[\tau \mid \Pr\left(t(\mathcal{A}, \mathcal{P}, G_n) \leq \tau\right) \geq 1 - \tfrac{1}{n}\right].$$

We can now express our research question formally:

Definition 2.2 (Gossip stopping problem)**.** *Given a graph G_n, a gossip algorithm $\mathcal{A}$, and a gossip protocol $\mathcal{P}$, the* gossip stopping problem *is to determine $E[t(\mathcal{A}, \mathcal{P}, G_n)]$ and $\hat{t}(\mathcal{A}, \mathcal{P}, G_n)$, the expected and high probability stopping times.*

In this chapter we consider $\mathcal{A} \in \{\texttt{PUSH}, \texttt{PULL}, \texttt{EXCHANGE}\}$ and $\mathcal{P}$ = *algebraic gossip*, so when these parameters and G_n are understood from the context, we denote the expected and high probability stopping times as $E[t]$ and $\hat{t}$, respectively. Moreover, we usually measure the stopping time in *rounds* (in order to compare between the two time models) where one round equals n consecutive timeslots. Thus, we define the expected number of rounds as $E[r] = E[t]/n$ and $\hat{r} = \hat{t}/n$ as the number of rounds by which all nodes complete the task with high probability.

For clarity, we first present our proofs for the *asynchronous* time model and the `EXCHANGE` algorithm. Then, in Section 2.6 we extend the results to the *synchronous* cases and `PUSH` and `PULL`.

2.3 Linear Bound on a Ring via Queuing Theory

Before proving the main results of Theorem 2.1 in the next section we prove in this section a bound on the specific case of a ring network. This is a simpler case to prove and understand, and will be used as a basis for the proof of the general result. A ring of size n is a connected cycle where each node has one left and one right neighbor.

Theorem 2.2. *For the asynchronous time model and the ring graph of size n, the stopping time (measured in rounds) of algebraic gossip is linear both in expectation and with high probability, i.e., $E[r] = \Theta(n)$ and $\hat{r} = \Theta(n)$.*

Proof. The idea of the proof is to reduce the problem of network coding on the ring graph to a simple system of queues and use Jackson's theorem for open networks to bound the time it takes *helpful messages* to cross the network.

To simplify our analysis, we cut the ring in an arbitrary place and get a path graph (without loss of generality, we assume that the leftmost node in the path is v_1 and the rightmost node is v_n), see Fig. 2.2 (a). It is clear that the stopping time of the algebraic gossip protocol will be larger in a path graph than in a ring graph.

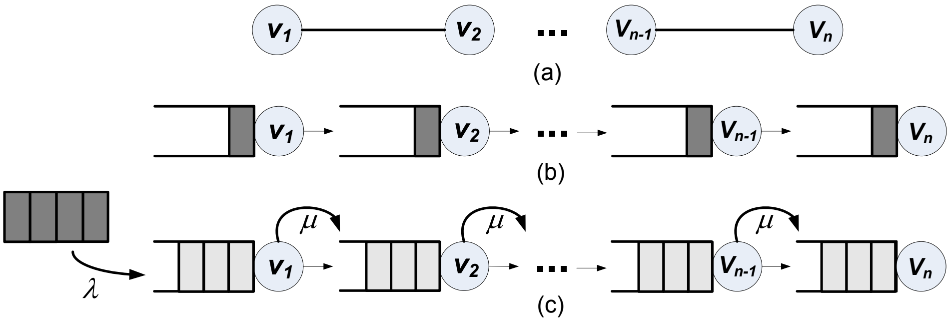


Figure 2.2: Modeling algebraic gossip in a path as a queuing network. (a) – Initial path graph. (b) – One real customer at each node. (c) – Queues are filled with dummy customers and real customers enter the system from outside.

Another simplification that we will do, for the first part of the proof, is to consider only the messages that travel from left to right (towards v_n) (i.e., other messages will be ignored, thus increasing the stopping time).

We define a queuing system by assuming a queue with a single server at each node. Customers of our queuing network are the *helpful messages*, i.e., messages that increase the rank of a node they arrive at. This means that every customer arriving at some node increases its rank by 1, so the queue size at a node represents a measure of *helpfulness* of the node to its right-hand neighbor (i.e., the queue size is the number of independent linear equations that the node can generate for its right-hand neighbor). The service procedure at node v_i is a transmission of a *helpful message* (customer) from v_i to v_{i+1}. So, from Lemma 2.1, the probability that a customer will be serviced at node v_i in a given timeslot is: $p \geq \frac{1}{n}(1 - \frac{1}{q})$, where $\frac{2}{n} \cdot \frac{1}{2} = \frac{1}{n}$ is the probability that in the `EXCHANGE` algorithm a message will be sent from v_i to v_{i+1} at any given timeslot.

Thus, we can consider that a service time in our queuing system is geometrically distributed with parameter p. The service time is distributed over the set $\{0, 1, 2, ...\}$, which means that a customer that enters an empty queue at the *end* of the timeslot can be immediately serviced with probability p (since it is the beginning of the next timeslot). A customer cannot pass more than one node (queue) in a single timeslot, so we define the transmission time as one timeslot. I.e., the time needed for a customer to pass through k queues is the sum of the waiting time in each queue, service time in each queue, and additional k timeslots for transmission from queue to queue.

The following lemma shows that the service rate can be bounded from below by an exponential random variable.

Lemma 2.2. *Let X be a geometric random variable with parameter p and supported on the set $\{0, 1, 2, \ldots\}$, i.e., for $k \in \mathbb{Z}^+$: $\Pr(X = k) = (1-p)^k p$, and let Y be an exponential random variable with parameter p. Then, for all $x \in \mathbb{R}^+$:*

$$\Pr(X \leq x) \geq \Pr(Y \leq x) = 1 - e^{-px}, \tag{2.1}$$

i.e., a random variable $Y \sim Exp(p)$ stochastically dominates the random variable $X \sim Geom(p)$.

Proof. For a geometric random variable X with a success probability p and supported on the set $\{0, 1, 2, 3, ...\}$:

$$\Pr(X > x) = (1-p)^{x+1}, \text{ for } x \in \mathbb{Z}^+$$

and

$$\Pr(X > x) = (1-p)^{\lfloor x \rfloor + 1}, \text{ for } x \in \mathbb{R}^+.$$

So, for $x \in \mathbb{R}^+$,

$$\Pr(X \leq x) = 1 - (1-p)^{\lfloor x \rfloor + 1} \geq 1 - (1-p)^x = 1 - e^{\ln(1-p)x}$$

and since $\ln(1-p) \leq -p$ we have: $\Pr(X \leq x) \geq 1 - e^{-px}$. Hence, if $Y \sim \text{Exp}(p)$ we obtain: $\Pr(X \leq x) \geq 1 - e^{-px} = \Pr(Y \leq x)$, i.e., random variable $Y \sim \text{Exp}(p)$ stochastically dominates the random variable $X \sim \text{Geom}(p)$. $\square$

We can now assume that the service time is exponentially distributed with parameter $\mu = p$. This assumption decreases the rate of transmission of *helpful messages*, and therefore will only increase the stopping time. The last is true since the probability that a customer will be serviced by time t_1 in a geometrical server is higher than in an exponential server, and thus each customer in a network with geometric servers will arrive at v_n by time t_2 with higher probability than in a network with exponential servers. The formal justification of this step is given later in Lemma 2.5, which proves this assertion for trees and not only for the line.

To this end, we have converted our network to a standard network of queues where the network is open, external arrivals to nodes will form a Poisson process, service times are exponentially distributed, and the queues are first come first serve (FCFS). For a queue i let μ_i denote the service rate and λ_i the total arrival rate. We present now Jackson's theorem for open networks; a proof of this theorem can be found in [19].

Jackson's Theorem. *In an open Jackson network of n queues where the utilization $\rho_i = \frac{\lambda_i}{\mu_i}$ is less than 1 at every queue, the equilibrium state probability distribution exists, and for state $(k_1, k_2, \ldots, k_n)$ is given by the product of the individual queue equilibrium distributions:* $\pi(k_1, k_2, ..., k_n) = \Pi_{i=1}^{n} \rho_i^{k_i}(1 - \rho_i)$.

We would like to use Jackson's theorem to conclude that there is an equilibrium state for our network of queues and that in the equilibrium state the lengths of the queues are independent. For Jackson's theorem to hold we need to appropriately define the arrival rate to the queues, so we will slightly change our queuing network.

The initial state of our system is that at every queue we have one *real* customer (see Fig. 2.2 (b)). Now we take all the n *real* customers out from the system and let them enter back via the leftmost queue with a predefined arrival rate. Clearly, this modification increases the stopping time. We define the *real* customers' arrivals as a Poisson process with rate $\lambda = \frac{\mu}{2}$. So, $\rho_i = \frac{\lambda_i}{\mu_i} = \frac{1}{2} < 1$ for all queues ($i \in [1..n]$).

Now, according to Jackson's theorem there exists an equilibrium state. So, our last step is to ensure that the lengths of all queues at time $t = 0$ are according to the equilibrium state probability distribution. We add *dummy* customers to all the queues according to the stationary distribution. By adding additional *dummy* customers (we call them *dummy* since their arrivals are not counted as a rank increment) to the system, we make the *real* customers wait longer in the queues, thus increasing the stopping time. Our queuing network with the above modifications is illustrated in Fig. 2.2 (c), where the *real* customers are dark, and the *dummy* customers are bright.

We will compute the stopping time in two phases. By the end of the first phase, node v_n will finish the algebraic gossip task. By the end of the second phase, all the nodes will finish the task. For the first phase, we find the time it takes the n'th (last) *real* customer to arrive at the rightmost node, i.e., node v_n. By that time, the rank of node v_n will become n and it will finish the algebraic gossip protocol (i.e., it received n *helpful messages*). Let us denote this time (in *timeslots*) as $t^{\overrightarrow{\text{arr}}} + t^{\overrightarrow{\text{cross}}}$, where $t^{\overrightarrow{\text{arr}}}$ is the time needed for the n'th customer to arrive at the first queue, and $t^{\overrightarrow{\text{cross}}}$ is the time needed for the n'th customer to pass through all the n queues in the system.

For the second phase, let us assume that after $t^{\overrightarrow{\text{arr}}} + t^{\overrightarrow{\text{cross}}}$ timeslots (when v_n finishes the algebraic gossip task) all nodes except node v_n forget all the information they have. So, the rank of all nodes except v_n is 0. Let us now analyze the information flow from the rightmost node in the path (v_n) to the leftmost node (v_1). In the same way, we will represent all *helpful messages* that node v_n will send as customers in our queuing system. In order to use Jackson's Theorem, we will again remove all the *real* customers from the system and will inject them to the queue of node v_n with a Poisson rate $\lambda = \mu/2$. We also fill all the queues in the system with *dummy* customers in order to achieve queue lengths that correspond to the equilibrium state distribution. Clearly, arrival of a *real* customer at some node v_i ($i \neq n$) will increase the rank of that node. So, after the last *real* customer arrives at node v_1, the ranks of *all* nodes will be n, and the algebraic gossip task will be finished.

Using the same equilibrium state analysis as before, we define the time it takes the n'th (last) *real* customer to arrive at the rightmost node v_n as $t^{\overleftarrow{\text{arr}}}$, and the time to cross all the n queues – arriving at node v_1 – as $t^{\overleftarrow{\text{cross}}}$.

So, $t = t^{\overrightarrow{\text{arr}}} + t^{\overrightarrow{\text{cross}}} + t^{\overleftarrow{\text{arr}}} + t^{\overleftarrow{\text{cross}}}$ is an upper bound for the number of timeslots needed to complete the task. Now we find the upper bound for $t^x, x \in \{\overrightarrow{\text{arr}}, \overrightarrow{\text{cross}}, \overleftarrow{\text{arr}}, \overleftarrow{\text{cross}}\}$ and then we will use union bound to obtain an upper bound on t.

From Jackson's Theorem, it follows that the number of customers in each queue is independent, which implies that the random variables that represent the waiting times in each queue are independent. To continue with the proof we need the following lemmas (the first is a classical result from queuing theory).

Lemma 2.3 ([49], section 4.3)**.** *Time needed to cross one $M/M/1$ queue (a queuing system in which interarrival and service times are distributed exponentially with parameters λ and μ, respectively) in the equilibrium state has an exponential distribution with parameter $\mu - \lambda$.*

Lemma 2.4. *Let Y be the sum of n independent and identically distributed exponential random variables (each with parameter $\mu > 0$) and $E[Y] = \frac{n}{\mu}$. Then, for $\alpha > 1$:*

$$\Pr(Y < \alpha E[Y]) > 1 - (2e^{-\alpha/2})^n. \tag{2.2}$$

Proof. Let $Y = \sum_{i=1}^{n} X_i$, where X_i are i.i.d. exponential random variables (each with parameter $\mu > 0$). The generating function of X is given by:

$$G_X(s) = \mathrm{E}\left[e^{sX}\right] = \int_0^\infty e^{sx} f_X(x)dx.$$

For any $s < \mu$: $G_X(s) = \frac{\mu}{\mu - s}$. Thus, the generating function of Y (sum of independent X_i's) for $s < \mu$: $G_Y(s) = (G_X(s))^n = \left(\frac{\mu}{\mu - s}\right)^n$. Now, we will apply a Chernoff bound on Y. For $\mu > s \geq 0$:

$$\Pr\left(Y \geq \alpha \mathrm{E}\left[Y\right]\right) = \Pr\left(Y \geq \alpha\frac{n}{\mu}\right)$$
$$= \Pr\left(e^{sY} \geq e^{s\cdot\alpha\frac{n}{\mu}}\right) \leq \frac{\mathrm{E}\left[e^{sY}\right]}{e^{s\cdot\alpha\frac{n}{\mu}}} = \frac{G_Y(s)}{e^{s\cdot\alpha\frac{n}{\mu}}}.$$

By letting $s = \mu/2$ we get:

$$\Pr\left(Y \geq \alpha \mathrm{E}\left[Y\right]\right) \leq \left(\frac{\mu}{(\mu - \frac{\mu}{2})e^{\alpha\frac{\mu}{2\mu}}}\right)^n = \left(2e^{-\alpha/2}\right)^n$$

and thus:

$$\Pr\left(Y < \alpha \mathrm{E}\left[Y\right]\right) > 1 - \left(2e^{-\alpha/2}\right)^n.$$

□

Recall that: $\mu = p \geq \frac{1}{n}(1 - \frac{1}{q})$ so $\mu \geq \frac{q-1}{qn} \geq \frac{1}{2n}$ for $q \geq 2$. The random variable $t^{\overrightarrow{\text{arr}}}$ is the sum of n independent random variables distributed exponentially with parameter $\mu/2$. From Lemma 2.3 we obtain that $t^{\overrightarrow{\text{cross}}}$ is the sum of n independent random variables distributed exponentially with parameter $\mu - \lambda = \frac{\mu}{2}$. It is clear that $t^{\overleftarrow{\text{arr}}}$ is distributed exactly as $t^{\overrightarrow{\text{arr}}}$ and $t^{\overleftarrow{\text{cross}}}$ is distributed exactly as $t^{\overrightarrow{\text{cross}}}$. Therefore (for $\mu = \frac{1}{2n}$): $\mathrm{E}\left[t^x\right] = \sum_{i=1}^n \frac{2}{\mu} = 4n^2$. Using Lemma 2.4 (with $\alpha = 2$) we obtain for $x \in \{\overrightarrow{\text{arr}}, \overrightarrow{\text{cross}}, \overleftarrow{\text{arr}}, \overleftarrow{\text{cross}}\}$:

$$\Pr\left(t^x \leq 8n^2\right) \geq 1 - \left(\tfrac{2}{e}\right)^n. \tag{2.3}$$

Using a union bound we get that:

$$\Pr\left(t \leq 32n^2\right) \geq \Pr\left(\cap_x t^x \leq 8n^2\right) \tag{2.4}$$
$$= 1 - \Pr\left(\cup_x t^x > 8n^2\right) \tag{2.5}$$
$$\geq 1 - 4\left(\tfrac{2}{e}\right)^n. \tag{2.6}$$

It is clear that $\Pr\left(t \leq 32n^2\right)$ increases when μ increases (faster server yields smaller waiting time); hence, the above inequality holds for any $\mu \geq \frac{1}{2n}$.

So, for the asynchronous time model and EXCHANGE we obtain an upper bound for the high probability stopping time: $\hat{t} = O(n^2)$ in timeslots, and thus $\hat{r} = O(n)$, in rounds. Let us now find an upper bound for the expected number of rounds needed to complete the task – $\mathrm{E}[r]$:

$$\mathrm{E}[r] = \tfrac{1}{n}\mathrm{E}[t] = \tfrac{1}{n}\mathrm{E}\left[t^{\overrightarrow{\text{arr}}} + t^{\overrightarrow{\text{cross}}} + t^{\overleftarrow{\text{arr}}} + t^{\overleftarrow{\text{cross}}}\right] \tag{2.7}$$

$$= \tfrac{4}{n}\mathrm{E}[t^x] = \tfrac{4}{n}4n^2 = 16n = O(n). \tag{2.8}$$

The lower bound is clear since in order to finish the algebraic gossip task each node has to receive at least n messages, so at least n^2 messages need to be sent and received. Since in each timeslot at most 2 messages (using EXCHANGE) are sent, we get: $\hat{t} = \Omega(n^2)$, thus $\hat{r} = \Omega(n)$, and $\mathrm{E}[r] = \Omega(n)$. The result of Theorem 2.2 is then follows: $E[r] = \Theta(n)$, and $\hat{r} = \Theta(n)$. □

2.4 Algebraic Gossip on Arbitrary Graphs

Now we are ready to prove our main results. First, we present the upper bound for any graph as a function of its maximum degree Δ, and then we give corollaries that are applications of this result for more specific cases.

Theorem 1 (restated). *For the asynchronous time model and for any graph G_n of size n with maximum degree Δ, the stopping time of algebraic gossip is $O(\Delta n)$ rounds both in expectation and with high probability.*

Proof. Consider an arbitrary graph G_n of size n with a maximum degree Δ and a vertex v. We pick any spanning tree rooted at v and will only consider messages that are sent from the tree edges towards v, i.e., we ignore all messages received from non-tree edges or in the opposite direction (see Fig. 2.3 (b)).

Now, let us concentrate on the information flow towards node v from all other nodes. As in the proof of Theorem 2.2, we will define a queuing system with a queue

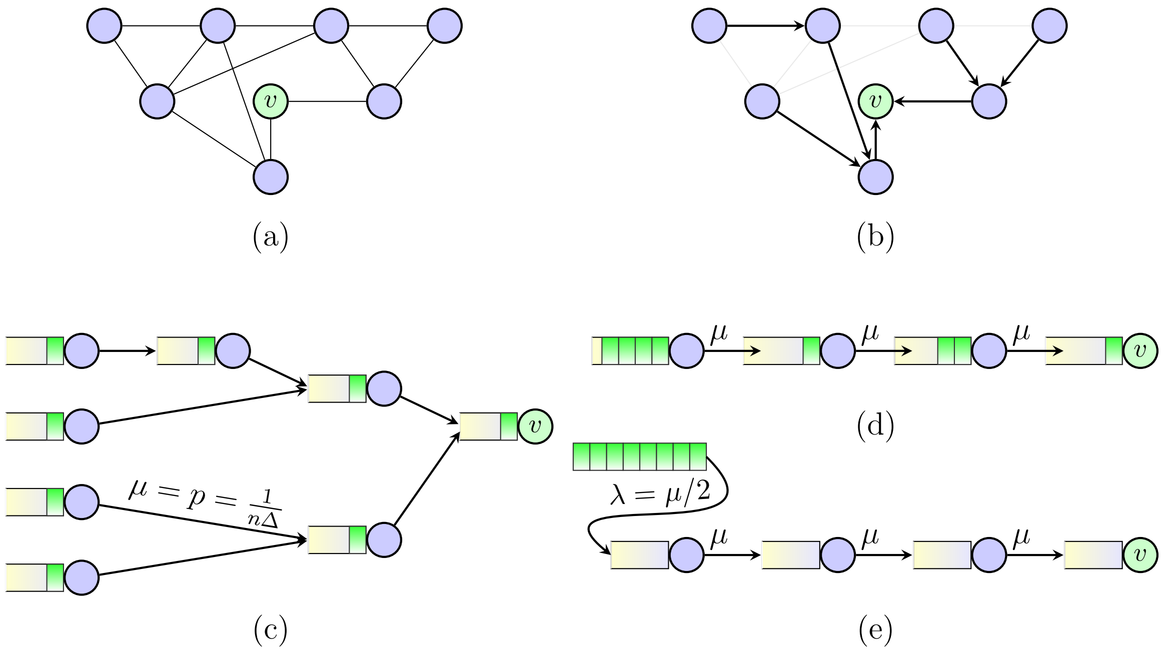


Figure 2.3: Reduction of Algebraic Gossip to a system of queues. (a) – Initial graph G. (b) – Spanning tree rooted at v, G_v. (c) – System of queues Q_n^{tree}. (d) – System of queues $Q_{l_{\max}}^{line}$. Stopping time of $Q_{l_{\max}}^{line}$ is larger than of Q_n^{tree}. (e)–Taking all customers out of the system and use Jackson theorem for open networks.

at each node (see Fig. 2.3 (c)). The following lemma shows that we can model the service time at each queue as an exponential random variable with parameter $\mu = p$.

Let T_n be a tree of size n rooted by node v. Let $\mathcal{N}(T_n, \mathcal{X})$ be a network of n queues where for each node u in T_n there is a queue and the queue output is connected to the input of the queue corresponding to the parent of u in T_n. In addition, each queue is of infinite size and initially has one customer in the queue (see Fig. 2.3) (c)). The servers of all the queues work with a service time distributed as $\mathcal{X}$. Let $t(T_n, \mathcal{X})$ be the random variable representing the time by which all the n customers in $\mathcal{N}(T_n, \mathcal{X})$ arrive to the queue of v (we assume v does not serve the customers).

Lemma 2.5. *For any tree* T_n *and* $0 < p \leq 1$*:*

$$\Pr\left(t(T_n, \mathrm{Geom}(p)) \leq \tau\right) \geq \Pr\left(t(T_n, \mathrm{Exp}(p)) \leq \tau\right), \text{ for all } \tau \geq 0.$$

Proof. We will prove this lemma by showing that for each customer c and for each queue u on the unique path that c traverses to the root, the probability that c reaches u before time τ is larger in $\mathcal{N}(T_n, \mathrm{Geom}(p))$ than in $\mathcal{N}(T_n, \mathrm{Exp}(p))$.

Consider a reverse topological order of the nodes in T_n (spanning tree of G_n rooted at v), $v^1, v^2, \ldots, v^n = v$, i.e., for every node v^i, $1 \leq i < n$, the parent of v^i is a node v^j and $j > i$. For a node v^i let C^i be the set of customers that it needs to serve on their way to the root. For a node v^i and a customer $c \in C^i$ let $\mathcal{G}_c^i(\tau)$ denote the event that c reached v^i before time τ in $\mathcal{N}(T_n, \mathrm{Geom}(p))$ and let $\mathcal{E}_c^i(\tau)$ be defined similarly for $\mathcal{N}(T_n, \mathrm{Exp}(p))$. We claim that for each $1 \leq i \leq n$, and each $c \in C^i$, $\Pr(\mathcal{G}_c^i(\tau)) \geq \Pr(\mathcal{E}_c^i(\tau))$, and the proof will be by induction on i.

Induction basis: $\Pr(\mathcal{G}_{v^1}^1(\tau)) \geq \Pr(\mathcal{E}_{v^1}^1(\tau))$. By definition, v_1 is a leaf with one customer, itself, and no children, so $\Pr(\mathcal{G}_{v^1}^1(\tau)) = \Pr(\mathcal{E}_{v^1}^1(\tau)) = 1$ for $\tau \geq 0$.

Induction step: Assume the claim is true for $1 \leq i < n-1$ and we will prove it is true for $i+1$. If v^{i+1} is a leaf, then we are done since this is an identical case to the base case. Assume v^{i+1} is not a leaf. The case $c = v^{i+1}$ is trivial so consider $c \in C^{i+1}$ that is not v^{i+1}. Then c must reach v^{i+1} via one of its children, let it be v^k where $k < i+1$. Then by the induction assumption $\Pr(\mathcal{G}_c^k(\tau')) \geq \Pr(\mathcal{E}_c^k(\tau'))$ for any τ', and from Lemma 2.2 for any τ the probability that a customer will be served by time τ is larger in $\mathcal{N}(T_n, \mathrm{Geom}(p))$ than in $\mathcal{N}(T_n, \mathrm{Exp}(p))$, so we have a faster arrival rate and a faster service rate and the lemma follows. □

The result of the above Lemma 2.5 is that any probabilistic upper bound on the stopping time of v in a tree network with exponential servers holds for the same tree network with geometric servers (both with the same parameter p and initially one customer at each queue).

Once all *real* customers arrive at v, it will reach rank n and will finish the algebraic gossip task. Now we have to calculate the service time parameter p. The degree of each node in G_n is at most Δ. Each node in T_n, except v, has a parent. Since we virtually remove (i.e., ignore) all edges that do not belong to T_n, at each node there is exactly one edge that goes towards the root v. Therefore, the probability that a customer will be serviced (transmitted towards v) at the end of a given timeslot is at least: $p \geq \left(\frac{2}{n} \cdot \frac{1}{\Delta}\right)\left(1 - \frac{1}{q}\right)$, where $\frac{2}{n} \cdot \frac{1}{\Delta} = \frac{2}{n\Delta}$ is the probability that in the `EXCHANGE` algorithm a message will be sent on the edge that goes towards v during one timeslot, and $(1-\frac{1}{q})$ is the minimal probability that the message is *helpful* (Lemma 2.1). Clearly, $p \geq \frac{1}{n\Delta}$ for $q \geq 2$, so we set our exponential servers to work with rate $\mu = \frac{1}{n\Delta}$.

Theorem 2.3. *Let Q_n^{tree} be a network of n nodes arranged in a tree topology, rooted at the node v. Each node has an infinite queue, and a single exponential server with parameter μ. Initially, there is a single customer in every queue. The time by which all n customers leave the network via the root node v is $t(Q_n^{tree}) = O(n/\mu)$ with high probability. Formally, for any $\alpha > 1$:*

$$\Pr\left(t(Q_n^{tree}) < \alpha 4n/\mu\right) > 1 - 2(2e^{-\alpha/2})^n. \tag{2.9}$$

The main idea of the Theorem 2.3 proof is to show that the stopping time of the network Q_n^{tree} (i.e., the time by which all the customers leave the network) is stochastically [3] smaller or equal to the stopping time of the systems of $l_{\max}$ queues arranged in a line topology – $Q_{l_{\max}}^{line}$ ($l_{\max}$ is the depth of the tree Q_n^{tree}). Then, we make the system $Q_{l_{\max}}^{line}$ stochastically slower by moving all the customers out and make them enter the system via the farthest queue with the rate $\lambda = \mu/2$. Finally, we use Jackson's Theorem for open networks (similar to the proof of Theorem 2.2) to find the stopping time of the system. See Fig. 2.3 for the illustration. The full proof of Theorem 2.3 can be found in Section 2.7.

[3] Stochastic dominance is formally defined in Section 2.7.

Using Theorem 2.3 for the tree T_n and with $\mu = \frac{1}{\Delta n}$, we obtain the stopping time of the node v: $t_v < \alpha 4n^2\Delta$ with probability of at least $1 - 2(2e^{-\alpha/2})^n$.

The same analysis holds for any node $u \in V$, i.e., we consider a spanning tree T_n rooted at u and find the stopping time of u, t_u. So, we can use a union bound to obtain the stopping time of all the nodes in G_n:

$$\Pr\left(\bigcap_{u\in V}\left(t_u < \alpha 4n^2\Delta\right)\right) \geq 1 - 2n\left(2e^{-\alpha/2}\right)^n. \tag{2.10}$$

By letting $\alpha = 2$ we obtain:

$$\Pr\left(\bigcap_{u\in V}\left(t_u < 8n^2\Delta\right)\right) \geq 1 - 2n\left(\tfrac{2}{e}\right)^n. \tag{2.11}$$

So, we determined that the stopping time of the algebraic gossip in G_n is $O(\Delta n^2)$ timeslots with high probability and thus: $\hat{r} = O(\Delta n)$.

The high probability bound of Eq. (2.10) is true for any $\alpha > 1$ and therefore strong enough to bound the expectation.

$$\Pr\left(t \geq 4\alpha n^2\Delta\right) \leq 2n(2e^{-\alpha/2})^n.$$

For a positive integer random variable t holds: $\mathrm{E}[t] = \sum_{i=1}^{\infty}\Pr(t \geq i)$. So, we have:

$$\begin{aligned}
\mathrm{E}[t] &= \sum_{i=1}^{\infty}\Pr(t \geq i) && (2.12)\\
&= \sum_{i=1}^{8n^2\Delta-1}\Pr(t \geq i) + \sum_{i=8n^2\Delta}^{\infty}\Pr(t \geq i) && (2.13)\\
&\leq 8n^2\Delta + \sum_{i=8n^2\Delta}^{\infty}\Pr(t \geq i) && (2.14)\\
&\leq 8n^2\Delta + 4n^2\Delta\sum_{\alpha=2}^{\infty}\Pr(t \geq 4\alpha n^2\Delta). && (2.15)
\end{aligned}$$

The last inequality is true since $\forall i \leq j, \Pr(t \geq i) \geq \Pr(t \geq j)$ and thus we can replace all $\Pr(t \geq i)$ for $i \in [4\alpha n^2\Delta, ..., 4(\alpha+1)n^2\Delta - 1]$ with $4n^2\Delta \times \Pr(t \geq 4\alpha n^2\Delta)$. Hence,

$$\mathrm{E}[t] \leq 8n^2\Delta + 4n^2\Delta \sum_{\alpha=2}^{\infty} \Pr(t \geq 4\alpha n^2\Delta) \tag{2.16}$$

$$\leq 8n^2\Delta + 4n^2\Delta \sum_{\alpha=2}^{\infty} 2n(2e^{-\frac{\alpha}{2}})^n \tag{2.17}$$

$$= 8n^2\Delta + 8n^3\Delta 2^n \sum_{\alpha=2}^{\infty} (e^{-n/2})^\alpha \tag{2.18}$$

$$= 8n^2\Delta + 8n^3\Delta 2^n \frac{e^{-n}}{1 - e^{-n/2}} \tag{2.19}$$

$$= 8n^2\Delta + \frac{8n^3\Delta}{1 - e^{-n/2}} \left(\frac{2}{e}\right)^n, \tag{2.20}$$

$$\text{for } n > 6 : \tag{2.21}$$

$$\leq 8n^2\Delta + 8n^2\Delta. \tag{2.22}$$

Now we can finish the proof of Theorem 2.1 by concluding:

$$E[t] = O(\Delta n^2) \text{ and } E[r] = O(\Delta n). \tag{2.23}$$

□

From Theorem 2.1, and since the maximum degree Δ is at most n, we can derive a general upper bound of algebraic gossip on any graph.

Corollary 2.1. *For the asynchronous time model and any graph G_n of size n, the gossip stopping time of the algebraic gossip task is $O(n^2)$ rounds, both in expectation and with high probability.*

We can use Theorem 2.1 to obtain a tight linear bound of algebraic gossip on graphs with a constant maximum degree. We note that previous bounds for this case are not tight, for example, for the ring graph the bound of [47] is $O(n^2)$.

Corollary 2.2. *For the asynchronous time model and any graph G_n of size n with a constant maximum degree Δ, the gossip stopping time of the algebraic gossip task is $O(n)$ rounds both in expectation and with high probability.*

We now show that the upper bound $O(\Delta n)$, presented in Theorem 2.1, is tight in the sense that for almost any Δ there exists a graph for which algebraic gossip takes $\Omega(\Delta n)$ rounds.

Theorem 2.4. *For any constant $\epsilon > 0$ and $2 \leq \Delta \leq (1-\epsilon)n$, and for the asynchronous time model there exists a graph G_n of size n with maximum degree Δ for which algebraic gossip takes $\Omega(\Delta n)$ rounds both in expectation and with high probability. In particular, there is a graph for which the stopping time is $\Omega(n^2)$ rounds both in expectation and with high probability.*

Proof. In order to prove this result we will need the following lemma.

Lemma 2.6. *Let X be a sum of m independent and identically distributed geometric random variables with parameter p, i.e., $X = \sum_{i=1}^{m} X_i$. Then, for any positive integer $k < m/p$*

$$\Pr(X > k) \geq 1 - \left(\frac{m}{e^{\frac{m-kp}{m}} kp} \right)^{-m}. \tag{2.24}$$

Proof. First, we will define Y as the sum of k independent Bernoulli random variables, i.e., $Y = \sum_{i=1}^{k} Y_i$, where $Y_i \sim Bernoulli(p)$. Let us notice that:

$$\Pr(X \leq k) = \Pr(Y \geq m)$$

The last is true since the event of observing at least m successes in a sequence of k Bernoulli trials implies that the sum of m independent geometric random variables is no more than k. From the other side, if the sum of m independent geometric random variables is no more than k it implies that m successes occurred not later than the k-th trial and thus $Y \geq m$.

Now we will use a Chernoff bound for the sum of independent Bernoulli random variables presented in [46]: For any $\delta > 0$ and $\mu = \mathrm{E}[Y]$:

$$\Pr(Y \geq (1+\delta)\mu) < \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}} \right)^{\mu}.$$

Since $\mu = \mathrm{E}[Y] = kp$, and by letting $\delta = \frac{m-kp}{kp}$, we obtain:

$$\Pr(Y \geq (1+\delta)\mu) = \Pr(Y \geq m) < \left(\frac{m}{e^{\frac{m-kp}{m}}kp}\right)^{-m}.$$

So:

$$\Pr(X \leq k) < \left(\frac{m}{e^{\frac{m-kp}{m}}kp}\right)^{-m},$$

and thus the result follows. □

Let us construct a graph G_n with $|V(G_n)| = n$ nodes and maximum degree $\Delta(G_n) = \Delta$. Consider two arbitrary graphs G' and G'' with certain maximum degrees $\Delta(G')$ and $\Delta(G'')$, respectively, and with total number of nodes n ($|V(G')|+|V(G'')| = n$). We now distinguish two cases: $\Delta \leq n/2$ and $\Delta > n/2$. For the first case ($\Delta \leq n/2$), let $u \in V(G')$ and $v \in V(G'')$, such that $d_u = \Delta(G') = \Delta - 1$ and $d_v = \Delta(G'') = \Delta - 1$. We construct G by interconnecting G' and G'' with a new edge (u, v), i.e., $V(G_n) = V(G') \cup V(G'')$ and $E(G_n) = E(G') \cup E(G'') \cup (u, v)$. See Fig. 2.4 (a) for an illustration.

For the second case ($\Delta > n/2$), the only difference in construction of G_n is the degree of $v \in V(G'')$, which is now $d_v = \Delta(G'') = n - \Delta - 1$.

In order to finish algebraic gossip on G, at least $\max\{|V(G')|, |V(G'')|\} \geq \frac{n}{2}$ messages should be sent over the edge (u, v). Using the fastest gossip variation – `EXCHANGE`, the probability p that a *helpful message* will be sent in one timeslot over the edge (u, v) is bounded by the probability that any message will be sent over (u, v), so: $p \leq \frac{1}{n}\left(\frac{1}{\Delta(G')+1} + \frac{1}{\Delta(G'')+1}\right)$.

For the first case ($\Delta \leq n/2$) we obtain:

$$p \leq \frac{1}{n}\left(\frac{1}{\Delta} + \frac{1}{\Delta}\right) = \frac{2}{n\Delta}. \tag{2.25}$$

For the second case we get:

$$p \leq \frac{1}{n}\left(\frac{1}{\Delta} + \frac{1}{n-\Delta}\right) = \frac{1}{\Delta(n-\Delta)} \leq \frac{1}{\Delta(n-(1-\epsilon)n)} = \frac{1}{n\Delta\epsilon}. \tag{2.26}$$

We can see that the first case can be viewed as the second with $\epsilon = 0.5$; thus, we can further analyze only the second case. The number of timeslots, t, needed to send $n/2$ *helpful messages* over the edge (u, v), can be viewed as a sum of $n/2$ geometric random variables with parameter p. Clearly, $\mathrm{E}[t] = \frac{n}{2} \cdot \frac{1}{p} = \frac{n^2\Delta\epsilon}{2} = \Omega(\Delta n^2)$ timeslots in both cases. Using Lemma 2.6 with $k = \lfloor \mathrm{E}[t]/2 \rfloor = \lfloor n^2\Delta\epsilon/4 \rfloor$, $p = \frac{1}{n\Delta\epsilon} \cdot \frac{n^2\Delta\epsilon/4}{\lfloor n^2\Delta\epsilon/4 \rfloor} \geq \frac{1}{n\Delta\epsilon}$ (we took p even larger than its maximum value; this will make calculations nicer and will not affect the bound), and $m = n/2$ we get:

$$\Pr\left(t > \lfloor n^2\Delta\epsilon/4 \rfloor\right) \geq 1 - \left(\frac{m}{e^{\frac{m-kp}{m}} kp}\right)^{-m} \tag{2.27}$$

$$= 1 - \left(\sqrt{e}/2\right)^{n/2}. \tag{2.28}$$

It is clear that $\Pr(t \geq k)$ increases when p decreases (the smaller probability of success – the larger the probability to finish later). Hence, the above inequality holds for any $p \leq \frac{1}{n\Delta\epsilon}$.

Thus, the number of timeslots needed is at least $\lfloor n^2\Delta\epsilon/4 \rfloor$ w.h.p. and $n^2\Delta\epsilon/2$ in expectation. So, the total stopping time of the algebraic gossip protocol on the graph G (measured in rounds) is: $\hat{r} = \Omega(\Delta n)$, and $\mathrm{E}[r] = \Omega(\Delta n)$, where $2 \leq \Delta \leq (1-\epsilon)n$ for any constant $\epsilon > 0$. The lower bound of $\Omega(n^2)$ rounds is achieved, for example, in a barbell graph – two cliques interconnected with a single edge (see Fig. 2.4 (b)). □

2.5 `EXCHANGE` Can Be Unboundedly Faster Than `PUSH` or `PULL`

As we presented earlier, there are three gossip variations: `PUSH`, `PULL`, and `EXCHANGE`. In `PUSH` or `PULL` there is only one message sent between the communication partners, in `EXCHANGE` two messages are sent. Thus, the total message complexity for the same number of communication rounds is doubled. We would like to know: Is the stop-

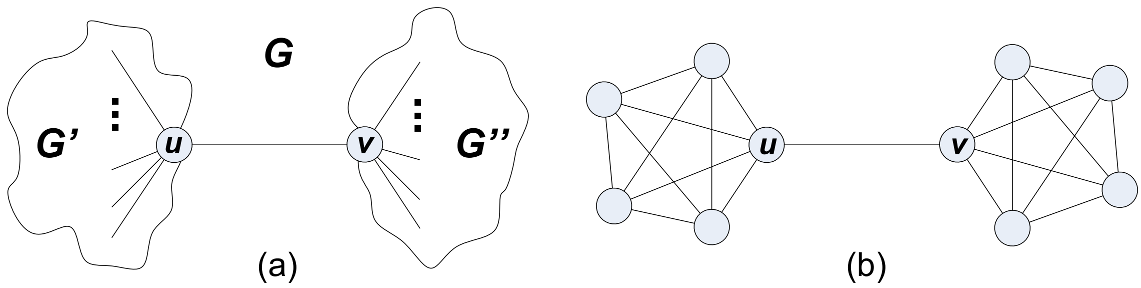


Figure 2.4: (a) Graph G_n, constructed from G' and G'', for the proof of Theorem 2.4. (b) An example of a G_n graph with $\Delta(G_n) = n/2$: barbell graph (two cliques of size $n/2$ connected with a single edge).

ping time decrease when using EXCHANGE worth the doubling message complexity? In this section we give the answer by presenting a graph for which the EXCHANGE gossip algorithm is *unboundedly* faster than the PUSH or PULL algorithms.

Theorem 2.5. *For the star graph S_n (which is a tree of n nodes with one node having degree $n-1$ and the other $n-1$ nodes having degree 1), algebraic gossip using* EXCHANGE *is unboundedly better than using* PUSH *or* PULL *algorithms. Formally, for $\mathcal{A} \in \{\texttt{PUSH}, \texttt{PULL}\}$:*

$$\lim_{n\to\infty} \frac{\hat{r}(\mathcal{A})}{\hat{r}(\texttt{EXCHANGE})} \to \infty, \quad \text{and} \quad \lim_{n\to\infty} \frac{E\left[r(\mathcal{A})\right]}{E\left[r(\texttt{EXCHANGE})\right]} \to \infty. \tag{2.29}$$

The proof of this theorem is a direct consequence of the following lemmas.

Lemma 2.7. *For the star graph S_n, algebraic gossip using **PUSH** takes $\Omega(n^2)$ rounds with high probability and in expectation.*

Proof. We are interested in a lower bound, so we will consider the minimum number of rounds to complete the task. The center node can finish the algorithm after one round since all other nodes will send (PUSH) to it their messages and in the best case all these messages will be *helpful*, so we ignore this phase. Now, the center node should send (PUSH) to every other node $n-1$ independent linear equations. In the synchronous time model, the center node wakes up exactly once in a round. Thus, the number of rounds needed to `PUSH` $n-1$ messages to all the $n-1$ other nodes is at least $(n-1)\cdot(n-1)$ with probability 1. In the asynchronous model, the center node will wake up in a given timeslot with probability $1/n$; thus, it will need $\Omega(n\cdot(n-1)\cdot(n-1))$ timeslots (to `PUSH` $n-1$ messages to all the $n-1$ other nodes) in expectation and with the high probability (sum of n independent geometric random variables). Thus, for both time models, the number of rounds needed is $\Omega(n^2)$. □

Lemma 2.8. *For the star graph S_n, algebraic gossip using* `PULL` *takes $\Omega(n\log n)$ rounds with high probability and in expectation.*

Proof. First, we give the following claim for the *coupon collector problem* [46].

Claim 2.1. *Let X be the r.v. for the number of coupons needed to obtain n distinct coupons (i.e., to obtain at least one coupon of each type), then:*

$$E[X] = \Theta(n\log n) \quad \textit{and w.h.p.} \quad X = \Theta(n\log n).$$

Proof. The first result (the expected value) and the upper bound w.h.p. are well known; see for example [48, 46]. We have not found a direct reference for the lower bound, namely that w.h.p. $X = \Omega(n\log n)$, so we give an outline here. Let $\mathcal{E}_x$ denote the event that all n different coupons have been collected after X steps. Let $X = \sum_{i=1}^{n} X_i$ where X_i is an r.v. that denotes the number of coupons of type i collected. Clearly X_i's are dependent. To overcome this difficulty we will use Poisson approximation of the binomial random variable X_i [46]. Consider n Poisson independent random variables Y_i ($i \in [1...n]$) with mean $\lambda = \frac{X}{n}$. Each variable represents the number of coupons of type i. Thus, the expected total number of coupons collected is X. Let $\mathcal{E}_y$ denote the Poisson version of the event $\mathcal{E}_x$, i.e., that after collecting the different types of coupons independently with Poisson distribution with λ, we have at least one type of each coupon. Since Y_i's are i.i.d., we have $\Pr(\mathcal{E}_y) = (\Pr(Y_i \geq 1))^n$.

It is clear that both $\Pr(\mathcal{E}_x)$ and $\Pr(\mathcal{E}_y)$ are monotonically increasing with X; therefore we can use the Poisson approximation that states that $\Pr(\mathcal{E}_x) \leq 2\Pr(\mathcal{E}_y)$ ([46], Corollary 5.11). Now, assume $X = n\ln n - n\ln\ln n$ and we have:

$$\Pr(\mathcal{E}_y) = \left(1 - e^{-(\ln n - \ln\ln n)}\right)^n = \left(1 - \frac{\ln n}{n}\right)^n.$$

Now we want to show that $\left(1 - \frac{\ln n}{n}\right)^n \leq \frac{1}{n}$. Let:

$$z = n\left(1 - \frac{\ln n}{n}\right)^n.$$

Then we obtain:

$$\ln z = \ln n + n\ln\left(1 - \frac{\ln n}{n}\right).$$

Using Taylor expansion we get:

$$\ln\left(1 - \frac{\ln n}{n}\right) \leq -\frac{\ln n}{n}.$$

So:

$$\ln z \leq \ln n - n\frac{\ln n}{n} = 0.$$

Since $\ln z \leq 0$ we get that $z \leq 1$ which yields: $\left(1 - \frac{\ln n}{n}\right)^n \leq \frac{1}{n}$. So, $\Pr(\mathcal{E}_x) \leq 2\Pr(\mathcal{E}_y) \leq \frac{2}{n}$ and thus:

$$\Pr(X \geq n\ln n - n\ln\ln n) = 1 - \frac{2}{n}.$$

□

The center node will finish the algorithm once it receives (PULL) a helpful message from every other node. Thus, the center node has to reach (PULL) every other node at least once. In the synchronous time model, the center node will transmit (wake up) exactly once in a round. Reaching every other node at least one time is exactly the coupon collector problem, so (using Claim 2.1): $\hat{r} = \Omega(n\log n)$, and $\mathrm{E}[r] =$

$\Omega(n \log n)$ rounds. In the asynchronous model, the center node will wake up in a given timeslot with probability $1/n$; thus, it needs $\Omega(n \cdot n \log n)$ timeslots in order to wake up $\Omega(n \log n)$ times in expectation and with high probability (lower bound on sum of i.i.d. geometric r.v.'s). □

Lemma 2.9. *For the star graph S_n, algebraic gossip using* **EXCHANGE** *takes $O(n)$ rounds with high probability and in expectation.*

Proof. To prove Lemma 2.9 we will use the following claim.

Claim 2.2. *Let X_i be independent geometric random variables with parameter p, and let $X = \sum_{i=1}^{n} X_i$. For $p \geq \frac{1}{2}$, and $\alpha > 1$:*

$$\Pr\left(X \geq 2n\alpha\right) \leq \left(2^{1.5-\alpha}\right)^n . \tag{2.30}$$

Proof. In order to obtain this upper bound on the sum of n independent geometric random variables we will use a Chernoff bound. The generating function of a geometric random variable X_i is given by:

$$G_{X_i}(t) = \mathrm{E}\left[e^{tX_i}\right] = \frac{pe^t}{1-(1-p)e^t} \text{ , where } t < -\ln(1-p).$$

The generating function of the sum of independent random variables is a multiplication of their generating functions. Thus:

$$G_X(t) = \mathrm{E}\left[e^{t\sum_{i=1}^{n} X_i}\right] = \mathrm{E}\left[e^{ntX_i}\right] = \left(\frac{pe^t}{1-(1-p)e^t}\right)^n .$$

Now, we will apply Markov's inequality to obtain an upper bound on X. For $t \geq 0$:

$$\Pr\left(X \geq 2n\alpha\right) = \Pr\left(e^{tX} \geq e^{t2n\alpha}\right) \leq \frac{\mathrm{E}\left[e^{tX}\right]}{e^{t2n\alpha}} = \frac{G_X(t)}{e^{t2n\alpha}}.$$

By letting $t = -0.5\ln(1-p)$ we get:

$$\Pr(X \geq 2n\alpha) \leq \left(\frac{(1-p)^{\alpha-0.5}p}{1-(1-p)^{0.5}}\right)^n .$$

It is clear that $\Pr(X \geq 2n\alpha)$ decreases when p increases. Thus, to obtain an upper bound, we will substitute p with its minimal value, i.e., $1/2$, and we get the result:

$$\begin{aligned}\Pr(X \geq 2n\alpha) &\leq \left(\frac{(1-0.5)^{\alpha-0.5}0.5}{1-(1-0.5)^{0.5}}\right)^n \\ &\leq \left(2 \cdot 0.5^{\alpha-0.5}\right)^n \\ &= \left(2^{1.5-\alpha}\right)^n .\end{aligned}$$

□

First, we consider the synchronous time model. Let us split the task into two phases. The first phase is the time (in rounds) r_1 until the center node v_1 learns all the initial messages, i.e., $dim(S_{v_1}(t)) = n$. The second phase is the time (in rounds) r_2 it takes v_1 to distribute the information to all the nodes.

Initially, every node $u \in V \setminus \{v_1\}$ is *helpful* to v_1. By Lemma 2.1, a message sent from u to v_1 will be *helpful* with probability of at least $1 - \frac{1}{q}$; thus, after n rounds, a node u will send a *helpful message* to v_1 with probability of at least $1 - \left(\frac{1}{2}\right)^n$ (for $q > 2$). Using union bound we can find the probability that all the nodes $u \in V \setminus \{v_1\}$ will send a helpful message to v_1 after n rounds:

$$\Pr(r_1 > n) \leq \sum_{u \in V \setminus \{v_1\}} \left(\frac{1}{2}\right)^n \leq n \left(\frac{1}{2}\right)^n . \tag{2.31}$$

Now, from the beginning of phase two, $dim(S_{v_1}) = n$ and hence the node v_1 will be *helpful* to every other node until the rank of that node becomes n. From Lemma 2.1, a message transmitted to some node from a node *helpful* to it, will increase its dimension with probability $p \geq 1 - \frac{1}{q}$.

Let us define X_i^u as the number of rounds needed for v_1 to increase the rank of some node $u \in V \setminus \{v_1\}$. It is clear that X_i^u has a geometric distribution with parameter p. We are interested to find $X^u = \sum_{i=1}^{n} X_i^u$, which represents the number of rounds by which the rank of node u will become n. Using Claim 2.2 (and the fact that for $q > 2$, $p = 1 - \frac{1}{q} > \frac{1}{2}$), we obtain for any $\alpha > 1$ that:

$$\Pr(X^u < 2\alpha n) \geq 1 - (2^{1.5-\alpha})^n. \tag{2.32}$$

Using union bound, we obtain the probability that ranks of all nodes will become n after $2\alpha n$ rounds:

$$\Pr\left(\cup_{u \in V \setminus \{v_1\}} X^u \geq 2\alpha n\right) \leq \sum_{u \in V \setminus \{v_1\}} \Pr(X^u \geq 2\alpha n) \tag{2.33}$$

$$\leq n \left(2^{1.5-\alpha}\right)^n, \tag{2.34}$$

and thus:

$$\Pr\left(\cap_{u \in V \setminus \{v_1\}} X^u < 2\alpha n\right) \geq 1 - n \left(2^{1.5-\alpha}\right)^n. \tag{2.35}$$

So,

$$\Pr(r_2 < 2\alpha n) \geq 1 - n \left(2^{1.5-\alpha}\right)^n, \tag{2.36}$$

and for $\alpha = 2$:

$$\Pr(r_2 < 4n) \geq 1 - n \left(\frac{1}{\sqrt{2}}\right)^n. \tag{2.37}$$

Combining the two phases together, i.e., $r \leq r_1 + r_2$, we have:

$$\Pr(r > 5n) \leq \Pr(r_1 \geq n) + \Pr(r_2 \geq 4n) \tag{2.38}$$

$$\leq n \left(\frac{1}{2}\right)^n + n \left(\frac{1}{\sqrt{2}}\right)^n \tag{2.39}$$

$$\leq 2n \left(\frac{1}{\sqrt{2}}\right)^n, \tag{2.40}$$

and thus: $\hat{r} = O(n)$.

Let us now find an upper bound for the expected number of rounds needed to complete the task, $E[r]$. Since $r \leq r_1 + r_2$, we get: $\mathrm{E}[r] \leq \mathrm{E}[r_1] + \mathrm{E}[r_2]$. During the first phase, each node $u \in V \setminus \{v_1\}$ will send a *helpful message* to v_1 with probability of at least $\frac{1}{2}$. Thus, $\mathrm{E}[r_1] \leq 2n$. The high probability bound of (2.36) allow us to show that for sufficient large n:

$$E[R_2] \leq 4n + 1. \tag{2.41}$$

In order to prove the Equation 2.41, we first rewrite the high probability result of Eq. (2.36) for r_2 with $\alpha > 1$:

$$\Pr(r_2 \geq 2n\alpha) \leq n(2^{1.5-\alpha})^n.$$

For a positive integer random variable r_2 holds: $\mathrm{E}[r_2] = \sum_{i=1}^{\infty} \Pr(r_2 \geq i)$. So, we have:

$$\begin{aligned}
\mathrm{E}[r_2] &= \sum_{i=1}^{\infty} \Pr(r_2 \geq i) \\
&= \sum_{i=1}^{4n-1} \Pr(r_2 \geq i) + \sum_{i=4n}^{\infty} \Pr(r_2 \geq i) \\
&\leq 4n + \sum_{i=4n}^{\infty} \Pr(r_2 \geq i) \\
&\leq 4n + 2n \sum_{\alpha=2}^{\infty} \Pr(r_2 \geq 2n\alpha).
\end{aligned}$$

The last inequality is true since $\forall i \leq j, \Pr(r_2 \geq i) \geq \Pr(r_2 \geq j)$ and thus we can replace all $\Pr(r_2 \geq i)$ for $i \in [2n\alpha, ..., 2n(\alpha+1)-1]$ with $2n \times \Pr(r_2 \geq 2n\alpha)$. Hence,

$$\begin{aligned}
\mathrm{E}[r_2] &\leq 4n + 2n \sum_{\alpha=2}^{\infty} \Pr(r_2 \geq 2n\alpha) \\
&\leq 4n + 2n \sum_{\alpha=2}^{\infty} n(2^{1.5-\alpha})^n \\
&= 4n + 2n^2 \cdot 2^{1.5n} \sum_{\alpha=2}^{\infty} (2^{-n})^{\alpha} \\
&= 4n + 2n^2 \cdot 2^{1.5n} \frac{2^{-2n}}{1-2^{-n}} \\
&= 4n + \frac{2n^2 \cdot 2^{0.5n}}{2^n - 1}, \\
&\leq 4n + 1 \quad \text{, for } n > 19.
\end{aligned}$$

Hence, we obtain: $\mathrm{E}[r] = O(n)$. In order to justify the result for the asynchronous time model (in which a node wakes up at a given timeslot with probability $1/n$), we notice that a node v_1 will wake up $O(n)$ times after at most $O(n^2)$ timeslots (or $O(n)$ rounds) with expectation and with high probability (sum of n i.i.d. geometric r.v.'s). Thus, the lemma holds for both time models. $\square$

Since for $\mathcal{A} \in \{\texttt{PUSH}, \texttt{PULL}\}$: $\hat{r}(\mathcal{A}) = \Omega(n \log n)$ and $\mathrm{E}[r(\mathcal{A})] = \Omega(n \log n)$ (Lemmas 2.7 and 2.8), and from Lemma 2.9: $\hat{r}(\texttt{EXCHANGE}) = O(n)$ and $\mathrm{E}[r(\texttt{EXCHANGE})] = O(n)$, we are ready to conclude that:

$$\lim_{n\to\infty} \frac{\hat{r}(\mathcal{A})}{\hat{r}(\texttt{EXCHANGE})} \to \infty, \quad \text{and} \quad \lim_{n\to\infty} \frac{E[r(\mathcal{A})]}{E[r(\texttt{EXCHANGE})]} \to \infty.$$

2.6 Algebraic Gossip with Synchronous Time Model, and with PUSH and PULL

In this section we give theorems and corollaries that extend the results presented in the chapter to both time models (synchronous and asynchronous), and to the three gossip algorithms (PUSH, PULL, and EXCHANGE).

The first theorem shows that the general upper bound for algebraic gossip also holds for the synchronous time model.

Theorem 2.6. *For the* ***synchronous*** *time model and for any graph G_n with maximum degree Δ, the stopping time of algebraic gossip is $O(\Delta n)$ rounds with high probability.*

Proof. The proof for the synchronous time model is almost the same as in the asynchronous case. The analysis will be done in *rounds* instead of *timeslots*. The probability that a customer will be serviced (transmitted towards v) at the end of a given *round* is at least: $p \geq \left(1 - (\frac{\Delta-1}{\Delta})^2\right)(1 - \frac{1}{q})$, where $\left(1 - (\frac{\Delta-1}{\Delta})^2\right) = \frac{2}{\Delta} - \frac{1}{\Delta^2}$ is the probability that in the EXCHANGE algorithm at least one message will be sent on a specific edge (in a specific direction) during one *round*, and $(1 - \frac{1}{q})$ is the minimal probability that the message is *helpful* (Lemma 2.1).

$$p \geq (\tfrac{2}{\Delta} - \tfrac{1}{\Delta^2})(1 - \tfrac{1}{q}) \geq \tfrac{1}{\Delta}(1 - \tfrac{1}{q}) \geq \tfrac{1}{2\Delta} \text{ for } q \geq 2.$$

If node i has received a message during a specific round from node j it will ignore the additional message that can arrive from the same node j at the same round (this can happen if i chooses j and j chooses i in the `EXCHANGE` gossip scheme in one round). Clearly, this assumption can only increase the stopping time since we ignore (possibly helpful) information.

t^x, $x \in \{\overrightarrow{\text{arr}}, \overrightarrow{\text{cross}}\}$ are measured now in *rounds* and not in *timeslots*. Since $\mu = p \geq \frac{1}{2\Delta}$, using Lemma 2.4 (with $\alpha = 2$ and $\mathrm{E}\left[t^x\right] = \frac{2n}{\mu} = 4n\Delta$), we obtain:

$$\Pr\left(t^x < 8n\Delta\right) > 1 - \left(\tfrac{2}{e}\right)^n \text{ , for } x \in \{\overrightarrow{\text{arr}}, \overrightarrow{\text{cross}}\}.$$

The rest of the proof is the same as in the asynchronous case and thus the result follows. □

The following theorem proves that the worst-case lower bound for algebraic gossip also holds for the synchronous time model.

Theorem 2.7. *For any constant $\epsilon > 0$ and $2 \leq \Delta \leq (1-\epsilon)n$, and for the synchronous time model, there exists a graph G_n of size n with maximum degree Δ for which algebraic gossip takes $\Omega(\Delta n)$ rounds both in expectation and with high probability. In particular, there is a graph for which the stopping time is $\Omega(n^2)$ rounds both in expectation and with high probability.*

Proof. The proof is almost the same as in Theorem 2.4. The analysis will be done in *rounds* instead of *timeslots*.

Using the fastest gossip variation – `EXCHANGE`, the probability p that a *helpful message* will be sent in one timeslot over the edge (u, v) can be bounded (using a union bound) as: $p \leq \frac{1}{\Delta(G')+1} + \frac{1}{\Delta(G'')+1}$.

For the first case ($\Delta \leq n/2$) we obtain: $p \leq \frac{2\Delta}{\Delta \cdot \Delta} = \frac{2}{\Delta}$. For the second case ($\Delta > n/2$) we get:

$$p \leq \tfrac{1}{\Delta} + \tfrac{1}{n-\Delta} = \tfrac{n}{\Delta(n-\Delta)} \leq \tfrac{n}{\Delta(n-(1-\epsilon)\Delta)} = \tfrac{1}{\Delta\epsilon}. \tag{2.42}$$

The first case can be viewed as the second with $\epsilon = 0.5$; thus, we can further analyze only the second case. The number of rounds, r, needed to to send $n/2$ *helpful* messages over the edge (v, u), can be viewed as a sum of $n/2$ geometric random variables with parameter p. Clearly, $\mathrm{E}[r] = \frac{n}{2} \cdot \frac{1}{p} = \frac{n\Delta(1-\alpha)}{4} = O(\Delta n)$ rounds. The number of rounds, r, needed to to send $n/2$ *helpful messages* over the edge (u, v), can be viewed as a sum of $n/2$ geometric random variables with parameter p. Clearly, $\mathrm{E}[r] = \frac{n}{2} \cdot \frac{1}{p} = \frac{n\Delta\epsilon}{2} = O(\Delta n)$ rounds in both cases. Using Lemma 2.6 with $k = \lfloor \mathrm{E}[r]/2 \rfloor = \lfloor n\Delta\epsilon/4 \rfloor$, $p = \frac{1}{\Delta\epsilon} \cdot \frac{n\Delta\epsilon/4}{\lfloor n\Delta\epsilon/4 \rfloor} \geq \frac{1}{\Delta\epsilon}$ (we took p even larger than its maximum value; this will make calculations nicer and will not affect the bound), and $m = n/2$ we get:

$$\Pr\left(r > \lfloor n\Delta\epsilon/4 \rfloor\right) \geq 1 - \left(\frac{m}{e^{\frac{m-kp}{m}} kp}\right)^{-m} = 1 - \left(\sqrt{e}/2\right)^{n/2}. \tag{2.43}$$

The rest of the proof is the same as in Theorem 2.4. □

The following corollary shows that our bounds (upper and lower) for algebraic gossip on general graphs hold also for the `PUSH` and `PULL` gossip algorithms.

Corollary 2.3. *The results of Theorems 2.1, 2.4, 2.6, and 2.7 hold also for* `PUSH` *and* `PULL` *gossip algorithms.*

Proof. By moving from the `EXCHANGE` to the `PUSH` or `PULL` gossip algorithms, we change only the probability of sending a *helpful message* on a specific (directed) edge, i.e., the service time at each node will change. Easy to see that this probability will be decreased by a factor of 2 (i.e., the service time will become twice as long). Clearly, such a reduction will not affect the asymptotic bounds that were achieved using Lemmas 2.4, and 2.6. □

2.7 Proof of Theorem 2.3

In this section we present the full proof of the main theorem used for the upper bound of the running time of algebraic gossip.

Theorem 2.3 (restated): *Let Q_n^{tree} be a network of n nodes arranged in a tree topology, rooted at the node v. Each node has an infinite queue, and a single exponential server with parameter μ. Initially, there is a single customer in every queue. The time by which all n customers leave the network via the root node v is $t(Q_n^{tree}) = O(n/\mu)$ with high probability. Formally, for any $\alpha > 1$:*

$$\Pr\left(t(Q_n^{tree}) < \alpha 4n/\mu\right) > 1 - 2(2e^{-\alpha/2})^n. \tag{2.44}$$

For the proof of this theorem we need the following auxiliary definitions, claims, and lemmas.

Stochastic Dominance

Definition 2.3 (Stochastic dominance, stochastic ordering [37, 32]). *We say that a random variable X is stochastically less than or equal to a random variable Y if and only if $\Pr(X \leq t) \geq \Pr(Y \leq t)$ for any $t \geq 0$, and such a relation is denoted as: $X \preceq Y$.*

Definition 2.4 (Stochastic equivalence). *We say that a random variable X is stochastically equivalent to a random variable Y if and only if $\Pr(X \leq t) = \Pr(Y \leq t)$ for any $t \geq 0$, and such a relation is denoted as: $X \approx Y$.*

Claim 2.3. *If for $i \in \{1, 2\}$, $X_i \preceq Y_i$, X_i are independent and Y_i are independent, then: $\max_i X_i \preceq \max_i Y_i$.*

Proof.

$$\Pr(\max_i X_i \leq t) = \bigcap_i \Pr(X_i \leq t) = \prod_i \Pr(X_i \leq t) \geq \prod_i \Pr(Y_i \leq t) = \Pr(\max_i Y_i \leq t).$$

Hence, $\max_i X_i \preceq \max_i Y_i$. $\square$

Claim 2.4. *If for $i \in \{1, 2\}$, $X_i \preceq Y_i$, X_i are independent and Y_i are independent, then: $\sum_i X_i \preceq \sum_i Y_i$.*

Proof.

$$\Pr(X_1 + X_2 \leq t) = \int_{-\infty}^{t} f_{X_1+X_2}(s)ds,$$

$$\text{where } f_{X_1+X_2}(s) = f_{X_1}(s) * f_{X_2}(s).$$

Thus:

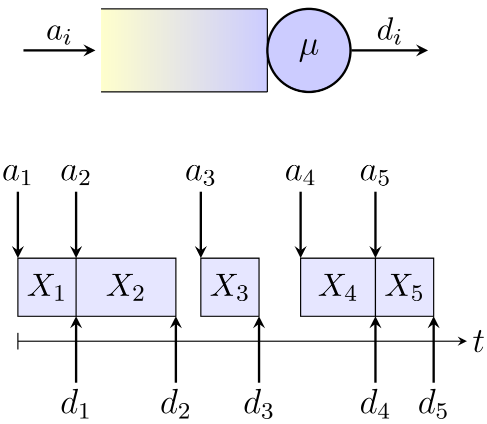
$$\begin{aligned}
\Pr(X_1 + X_2 \leq t) &= \int_{-\infty}^{t}\int_{-\infty}^{\infty} f_{X_1}(\tau)f_{X_2}(s-\tau)d\tau ds = \int_{-\infty}^{\infty} f_{X_1}(\tau)\Pr(X_2 \leq t-\tau)d\tau \\
&\geq \int_{-\infty}^{\infty} f_{X_1}(\tau)\Pr(Y_2 \leq t-\tau)d\tau = \int_{-\infty}^{t}\int_{-\infty}^{\infty} f_{X_1}(\tau)f_{Y_2}(s-\tau)d\tau ds \\
&= \int_{-\infty}^{t}\int_{-\infty}^{\infty} f_{Y_2}(\tau)f_{X_1}(s-\tau)d\tau ds = \int_{-\infty}^{\infty} f_{Y_2}(\tau)\Pr(X_1 \leq t-\tau)d\tau \\
&\geq \int_{-\infty}^{\infty} f_{Y_2}(\tau)\Pr(Y_1 \leq t-\tau)d\tau = \int_{-\infty}^{t}\int_{-\infty}^{\infty} f_{Y_2}(\tau)f_{Y_1}(s-\tau)d\tau ds \\
&= \Pr(Y_1 + Y_2 \leq t).
\end{aligned}$$

Hence, $\sum_{i=1}^{2} X_i \preceq \sum_{i=1}^{2} Y_i$. $\square$

Later arrivals yield later departures

Consider an infinite FCFS queue with a single exponential server. We define a_i as the time of arrival number i to the queue, and d_i as time of departure number i from the queue. Let X_i be the exponential random variable representing the service time of the arrival i. For all i, X_i's are *i.i.d.*

Let a_i be a sequence of m arrival times to the queue, and d_i be a sequence of m departure times from the queue (see Figure 2.5 for an illustration).



$$d_i = \max(a_i, d_{i-1}) + X_i$$

Figure 2.5: Arrival and departure times.

Lemma 2.10. *If the sequence a_i is replaced with another sequence of m arrivals – $\hat{a}_i$, such that: $\hat{a}_i \succeq a_i\ \forall i \in [1, ..., m]$, then the resulting sequence of m departures will be such that: $\hat{d}_i \succeq d_i\ \forall i \in [1, ..., m]$. I.e., if every new arrival occurred, stochastically, at the same time or later than the old arrival, then every new departure from the queue will occur, stochastically, at the same time or later than the old departure.*

Proof. The proof is by induction on the arrival index j, $j \in [1, ..., m]$.

- Induction basis: $\hat{d}_1 \succeq d_1$ follows since $d_1 = a_1 + X_1$, $\hat{d}_1 = \hat{a}_1 + X_1$, and $\hat{a}_1 \succeq a_1$.
- Induction assumption: $\forall i < j : \hat{d}_i \succeq d_i$.
- Induction step: we need to show that $\hat{d}_j \succeq d_j$.

If the j's arrival occurred when the server was busy, then $d_j = d_{j-1} + X_j$. If the server was idle when the j's arrival occurred, then $d_j = a_j + X_j$. Thus, we can write:

$$d_j = \max(d_{j-1}, a_j) + X_j, \tag{2.45}$$

$$\text{and } \hat{d}_j = \max(\hat{d}_{j-1}, \hat{a}_j) + X_j. \tag{2.46}$$

Since from induction assumption $\hat{d}_{j-1} \succeq d_{j-1}$, and $\hat{a}_j \succeq a_j$, using Claims 2.3 and 2.4 we obtain $\hat{d}_j \succeq d_j$. □

Proof of Theorem 2.3. We denote the nodes of the queuing system Q_n^{tree} as Z_j^l, where l ($l \in [1, ..., l_{\max}]$) is the level of the node in the tree, and j is the node's index in level l. The root of the Q_n^{tree} tree is the node Z_1^1. All servers in the Q_n^{tree} network are ON all the time (work-conserving scheduling), i.e., servers work whenever they have customers to serve. There are no external arrivals to the system. Once a customer is serviced on level l, it enters the appropriate queue at the level $l-1$. When a customer is serviced by the root Z_1^1, it leaves the network.

Now, let us define the auxiliary queuing systems: $\hat{Q}_n^{tree}$ and $Q_{l_{\max}}^{line}$.

Definition 2.5 (Network of queues $\hat{Q}_n^{tree}$)**.** *$\hat{Q}_n^{tree}$ is the same network as Q_n^{tree} with the following change in the servers' scheduling:*

At any given moment, only one server at every level l ($l \in [1, ..., l_{\max}]$) is ON. Once a customer leaves level l, a server that will be scheduled (turned ON) at level l, is the server that has in its queue a customer that has earliest arrival time to a queue at level l among all the current customers at level l. If there are customers that initially reside at level l, they will be serviced in the order of their IDs (we assume for analysis that every customer has a unique identification number).

Definition 2.6 (Network of queues $Q_{l_{\max}}^{line}$)**.** *$Q_{l_{\max}}^{line}$ is the following modification of the network Q_n^{tree} that results in a network of $l_{\max}$ queues arranged in a line topology.*

For all $l \in [1, .., l_{\max}]$, we merge all the nodes at the level l to a single node (a single queue with a single server). We name this single node at level l as the first node in Q_n^{tree} at level l, i.e., Z_1^l. The customers that initially reside at level l will be placed in a single queue in the order of their IDs. This modification results in $Q_{l_{\max}}^{line}$ – a network of $l_{\max}$ queues arranged in a line topology: $Z_1^{l_{\max}} \to Z_1^{l_{\max}-1} \to \cdots \to Z_1^1$.

Definition 2.7 (Network of queues $\grave{Q}_{l_{\max}}^{line}$)**.** *$\grave{Q}_{l_{\max}}^{line}$ – is the same system as $Q_{l_{\max}}^{line}$ with the following modification. We take the last customer at some node Z_1^m ($m \in [1, .., l_{\max} - 1]$) and place it at the head of the queue of node Z_1^{m+1}. I.e., we move one customer, one queue backward in the line of queues.*

Definition 2.8 (Network of queues $\hat{Q}^{line}_{l_{\max}}$). $\hat{Q}^{line}_{l_{\max}}$ *– is the same system as* $Q^{line}_{l_{\max}}$ *with the following modification. We move all the customers to queue* $Z_1^{l_{\max}}$. *I.e., all the customers have to traverse now through all the* $l_{\max}$ *queues in the line.*

We summarize the queuing systems defined above in short Table 2.1.

Q^{tree}_n	Original system of n queues arranged in a tree topology. Fig. 2.6 (a).
$\hat{Q}^{tree}_n$	System of n queues arranged in a tree topology. Only one server is active at each level at a given time. Fig. 2.6 (b).
$Q^{line}_{l_{\max}}$	System of $l_{\max}$ queues arranged in a line topology. Fig. 2.6 (c).
$\grave{Q}^{line}_{l_{\max}}$	System of $l_{\max}$ queues arranged in a line topology. One customer is moved one queue backward.
$\hat{Q}^{line}_{l_{\max}}$	System of $l_{\max}$ queues arranged in a line topology. All customers are moved backward to the queue $Z_1^{l_{\max}}$.

Table 2.1: Queuing systems used in the proof.

The proof of Theorem 2.3 consists of showing the following relations between the stopping times of the queuing systems:

$$t(Q^{tree}_n) \preceq t(\hat{Q}^{tree}_n) \approx t(Q^{line}_{l_{\max}}) \preceq t(\grave{Q}^{line}_{l_{\max}}) \preceq t(\hat{Q}^{line}_{l_{\max}}) = O(n/\mu). \tag{2.47}$$

Stopping time of a queuing system $t(Q)$ is the time at which the last customer leaves the system (via the node Z_1^1). In order to compare the stopping times of queuing systems, we define the following ordered set (or sequence) of departure time from a server Z in a queuing system Q: $d(Z,Q) = (d_1(Z,Q), d_2(Z,Q), ..., d_i(Z,Q), ...)$, where $d_i(Z,Q)$ is the time of the departure number i from the node (server) Z.

First, we want to show that the stopping time of Q^{tree}_n is at most the stopping time of the system $\hat{Q}^{tree}_n$, i.e., $t(Q^{tree}_n) \preceq t(\hat{Q}^{tree}_n)$.

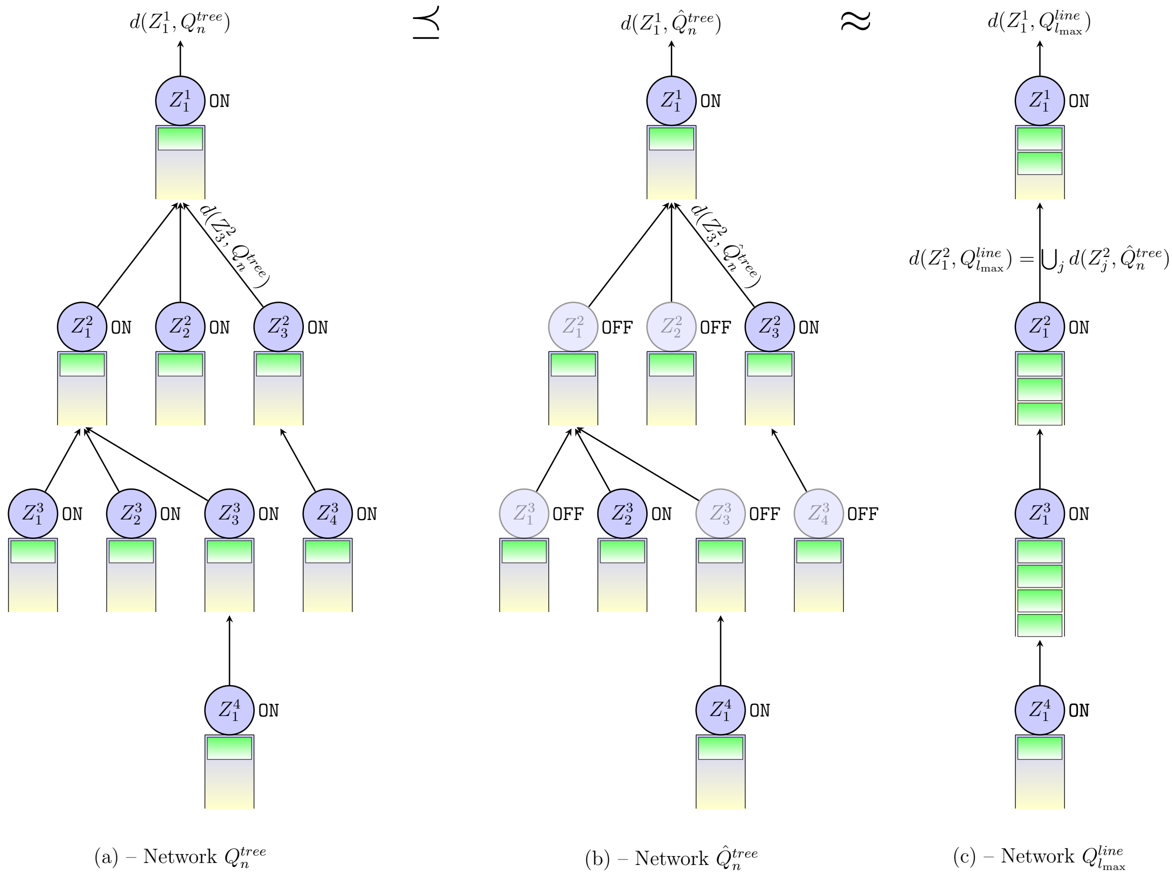


Figure 2.6: (a) Network Q_n^{tree}, where all the servers work all the time. (b) Network $\hat{Q}_n^{tree}$, where only one server at each level works at a given time. (c) Network $Q_{l_{\max}}^{line}$.

Lemma 2.11. *In $\hat{Q}_n^{tree}$, every departure from the system (via Z_1^1) will occur, stochastically, at the same time or later than in Q_n^{tree}:*

$$d_i(Z_1^1, \hat{Q}_n^{tree}) \succeq d_i(Z_1^1, Q_n^{tree}) \ \forall i \in [1, ..., n]. \tag{2.48}$$

Thus, in $\hat{Q}_n^{tree}$, the last customer will leave the system, stochastically, at the same time or later than in Q_n^{tree} or: $t(Q_n^{tree}) \preceq t(\hat{Q}_n^{tree})$.

Proof. The proof is by induction on the tree level l, $l \in [1, ..., l_{\max}]$.

- Induction basis: $\forall i, j : d_i(Z_j^{l_{\max}}, \hat{Q}_n^{tree}) \succeq d_i(Z_j^{l_{\max}}, Q_n^{tree})$. This is true since in $\hat{Q}_n^{tree}$, the nodes do not work all the time, and thus the departures will occur, stochastically, at the same time or later than in Q_n^{tree}. If there is a single node at level $l_{\max}$, in $\hat{Q}_n^{tree}$ it will be ON all the time as in Q_n^{tree}, and thus, the departures will occur, stochastically, at the same time in both systems.

- Induction assumption: for all $l > m$ ($m \geq 1$), $\forall i, j \,:\, d_i(Z_j^l, \hat{Q}_n^{tree}) \succeq d_i(Z_j^l, Q_n^{tree})$.
- Induction step: we need to show that: $\forall i, j \,:\, d_i(Z_j^m, \hat{Q}_n^{tree}) \succeq d_i(Z_j^m, Q_n^{tree})$.

By induction assumption, for $l = m+1$: $\forall i, j \,:\, d_i(Z_j^{m+1}, \hat{Q}_n^{tree}) \succeq d_i(Z_j^{m+1}, Q_n^{tree})$. Now let us take a look at the departures from a node Z_j^m. There are two cases: Z_j^m is a leaf, and Z_j^m is not a leaf. If Z_j^m is a leaf, we can use the same argument as in the induction basis: in $\hat{Q}_n^{tree}$, the node Z_j^m does not work all the time, and thus the departures from it in $\hat{Q}_n^{tree}$ cannot occur earlier than in Q_n^{tree}. If Z_j^m is not a leaf, it has input/inputs of arrivals from the level $m+1$. Since the arrivals from the level $m+1$ in $\hat{Q}_n^{tree}$ occur, stochastically, at the same time or later than in Q_n^{tree} (by induction assumption), even if node Z_j^m would work all the time (as in Q_n^{tree}), we would obtain from Lemma 2.10: $\forall i, j \,:\, d_i(Z_j^m, \hat{Q}_n^{tree}) \succeq d_i(Z_j^m, Q_n^{tree})$. Moreover, in $\hat{Q}_n^{tree}$, node Z_j^m does not work all the time (unless it is the only node at level m); thus the departure times in $\hat{Q}_n^{tree}$ can be even larger. □

Lemma 2.12. *In $Q_{l_{\max}}^{line}$, every departure from the system (via Z_1^1) will occur, stochastically, at the same time as in $\hat{Q}_n^{tree}$. Thus, in $Q_{l_{\max}}^{line}$, the last customer will leave the system, stochastically, at the same time as in $\hat{Q}_n^{tree}$.*

Proof. Consider the two following facts regarding the network $\hat{Q}_n^{tree}$. First, a customer entering level l will be serviced after all the customers that arrived at level l before it, are serviced. Second, at any given moment, only one customer is being serviced at level l (if there is at least one customer at the nodes Z_j^l). These facts are true due to the scheduling of the servers in $\hat{Q}_n^{tree}$ (Definition 2.5).

Clearly, the same facts are true for the network $Q_{l_{\max}}^{line}$. First, any customer entering level l will be serviced after all the customers that arrived at level l before it are serviced. Second, at any given moment, only one customer is being serviced at level l (if there is at least one customer in node Z_1^l). These facts are true since in $Q_{l_{\max}}^{line}$, at every level, there is a single queue with a single server (Definition 2.6).

So, the departure times of every customer from every level l ($l \in [1, ..., l_{\max}]$) are, stochastically, the same in both systems. The departures from level $l = 1$ are the departures from the node Z_1^1, and thus the lemma holds. $\square$

Now we are going to move one customer one queue backward, and will show that the resulting system will have stochastically larger (or the same) stopping time.

Lemma 2.13. *Consider a network $Q_{l_{\max}}^{line}$. Let m be a level index: $m \in [1, .., l_{\max} - 1]$. We take the last customer at node Z_1^m and place it at the head of the queue of node Z_1^{m+1}, and call the resulting network – $\grave{Q}_{l_{\max}}^{line}$ (Fig. 2.7 (b)). Then:*

$$d_i(Z_1^1, Q_{l_{\max}}^{line}) \preceq d_i(Z_1^1, \grave{Q}_{l_{\max}}^{line}) \ \forall i \in [1, ..., n]. \tag{2.49}$$

Thus, in $\grave{Q}_{l_{\max}}^{line}$, the last customer will leave the system, stochastically, at the same time or later than in $Q_{l_{\max}}^{line}$, or: $t(Q_{l_{\max}}^{line}) \preceq t(\grave{Q}_{l_{\max}}^{line})$.

Proof. We call the customer that was moved – customer c. Let us take a look at the times of arrivals to node Z_1^m in $Q_{l_{\max}}^{line}$ and in $\grave{Q}_{l_{\max}}^{line}$. Since customer c is already located in the queue of Z_1^m in $Q_{l_{\max}}^{line}$, its arrival time can be considered as 0. In $\grave{Q}_{l_{\max}}^{line}$, the arrival time of c is at least 0 (it should be serviced at Z_1^{m+1} before arriving at Z_1^m). Each of the rest of the customers that should arrive at Z_1^m will arrive in $\grave{Q}_{l_{\max}}^{line}$, stochastically, at the same time or later than in $Q_{l_{\max}}^{line}$, since in $\grave{Q}_{l_{\max}}^{line}$ the server Z_1^{m+1} should first service the customer c, and only then start servicing the rest customers. Thus, $d_i(Z_1^{m+1}, \grave{Q}_{l_{\max}}^{line}) \succeq d_i(Z_1^{m+1}, Q_{l_{\max}}^{line})$. Using Lemma 2.10 we obtain that: $d_i(Z_1^m, \grave{Q}_{l_{\max}}^{line}) \succeq d_i(Z_1^m, Q_{l_{\max}}^{line})$. Iteratively applying Lemma 2.10 to the nodes Z_1^l, $l \in [m-1, ..., 1]$, we obtain the result: $d_i(Z_1^1, \grave{Q}_{l_{\max}}^{line}) \succeq d_i(Z_1^1, Q_{l_{\max}}^{line})$. $\square$

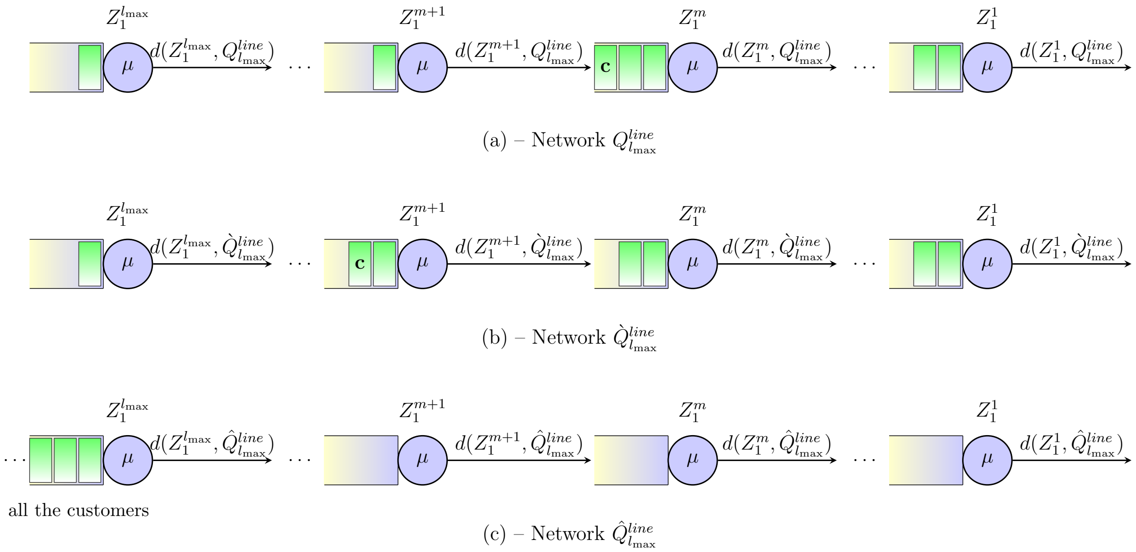


Figure 2.7: (a) Network $Q_{l_{\max}}^{line}$. (b) Network $\grave{Q}_{n}^{line}$, where one customer is moved one queue backward. (c) Network $\hat{Q}_{l_{\max}}^{line}$, where all the customers are at the last queue.

Corollary 2.4. *Consider a network $\hat{Q}_{l_{\max}}^{line}$ (Definition 2.8) that is identical to the network $Q_{l_{\max}}^{line}$ with the following change. In $\hat{Q}_{l_{\max}}^{line}$, all n customers are located at the node $Z_1^{l_{\max}}$ (Fig. 2.7 (c)). Then:*

$$d_i(Z_1^1, Q_{l_{\max}}^{line}) \preceq d_i(Z_1^1, \hat{Q}_{l_{\max}}^{line}) \ \forall i \in [1, ..., n]. \tag{2.50}$$

Thus, in $\hat{Q}_{l_{\max}}^{line}$, the last customer will leave the system, stochastically, at the same time or later than in $Q_{l_{\max}}^{line}$, or: $t(Q_{l_{\max}}^{line}) \preceq t(\hat{Q}_{l_{\max}}^{line})$.

Proof. Given the network $Q_{l_{\max}}^{line}$, we take one customer from the tail of some queue (except the queue of node $Z_1^{l_{\max}}$) and place it at the head of the queue of the preceding node in the $Q_{l_{\max}}^{line}$. According to Lemma 2.13, we get a network in which every customer leaves via Z_1^1, stochastically, not earlier than in $Q_{l_{\max}}^{line}$. Iteratively moving customers (one customer and one queue at a time) backwards we finally get the network $\hat{Q}_{l_{\max}}^{line}$ in which all n customers are located at node $Z_1^{l_{\max}}$. Since at each step, according to Lemma 2.13, the departure times from Z_1^1 could only get, stochastically, larger, the lemma holds. □

Corollary 2.5. *The time it will take the last customer to leave the network of n queues arranged in a* tree *topology is, stochastically, the same or smaller than in the network of n queues arranged in a* line *topology where all n customers are located at the farthest queue, i.e., $t(Q_n^{tree}) \preceq t(\hat{Q}_{l_{\max}}^{line})$.*

Proof. This result directly follows from Lemmas 2.11, 2.12, and the Corollary 2.4. $\square$

Now we are ready for the last step of the proof. We find the stopping time of a system of queues arranged in a line, with all the customers located at the last queue.

Lemma 2.14. *The time it will take for the last customer to leave system $\hat{Q}_{l_{\max}}^{line}$ ($l_{\max}$ queues arranged in a line) is $O(n/\mu)$ with high probability. Formally, for any $\alpha > 1$:*

$$\Pr\left(t(\hat{Q}_{l_{\max}}^{line}) < \alpha 4n/\mu\right) > 1 - 2(2e^{-\alpha/2})^n. \tag{2.51}$$

Proof. Initially, all the customers (from now we will call them *real* customers) are located in the last ($Z_1^{l_{\max}}$) queue. We now take all the *real* customers out of this queue and will make them enter the system (via $Z_1^{l_{\max}}$) from outside. We define the *real* customers' arrivals as a Poisson process with rate $\lambda = \frac{\mu}{2}$. So, $\rho = \frac{\lambda}{\mu} = \frac{1}{2} < 1$ for all the queues in the system. Clearly, such an assumption only increases the stopping time of the system (stopping time is the time until the last customer leaves the system). According to Jackson's theorem, whose proof can be found in [20], there exists an equilibrium state. So, we need to ensure that the lengths of all queues at time $t = 0$ are according to the equilibrium state probability distribution. We add *dummy* customers to all the queues according to the stationary distribution. By adding additional *dummy* customers to the system, we make the *real* customers wait longer in the queues, thus increasing the stopping time.

We will compute the stopping time $t(\hat{Q}_{l_{\max}}^{line})$ in two phases: Let us denote this time as $t_1 + t_2$, where t_1 is the time needed for the n'th customer to arrive at the first queue, and t_2 is the time needed for the n'th customer to pass through all the $l_{\max}$ queues in the system. From Jackson's Theorem, it follows that the number of customers in each queue is independent, which implies that the random variables that represent the waiting times in each queue are independent.

The random variable t_1 is the sum of n independent random variables distributed exponentially with parameter $\frac{\mu}{2}$. From Lem. 2.3 we obtain that t_2 is the sum of $l_{\max}$ independent random variables distributed exponentially with parameter $\mu - \lambda = \frac{\mu}{2}$. Since $l_{max} \leq n$, we can assume the worst case (for the upper bound of stopping time) $l_{max} = n$. Thus, we can view t_2 as the sum of n independent random variables distributed exponentially with parameter $\frac{\mu}{2}$. $\mathrm{E}[t_1] = \sum_{i=1}^{n} \frac{2}{\mu} = \frac{2n}{\mu}$, so, using Lem. 2.4:

$$\Pr(t_1 < \alpha \mathrm{E}[t_1]) > 1 - (2e^{-\alpha/2})^n, \tag{2.52}$$

$$\Pr(t_1 < \alpha 2n/\mu) > 1 - (2e^{-\alpha/2})^n. \tag{2.53}$$

In a similar way we obtain: $\Pr(t_2 < \alpha 2n/\mu) > 1 - (2e^{-\alpha/2})^n$. Since $t(\hat{Q}^{line}_{l_{\max}}) = t_1 + t_2$, using union bound, we obtain:

$$\Pr(t_1 + t_2 < \alpha 4n/\mu) > 1 - 2(2e^{-\alpha/2})^n. \tag{2.54}$$

So, for a constant α, $t(\hat{Q}^{line}_{l_{\max}}) = O(n/\mu)$, w.p. of at least $1 - 2(2e^{-\alpha/2})^n$. □

From Corollary 2.5 we have that $t(Q^{tree}_n) \preceq t(\hat{Q}^{line}_{l_{\max}})$ and thus: $t(Q^{tree}_n) < \alpha 4n/\mu$ w.p. of at least $1-2(2e^{-\alpha/2})^n$ for any $\alpha > 1$, so the proof of Theorem 2.3 is completed.

□

2.8 Conclusions

In this chapter we prove bounds on the stopping time of the algebraic gossip protocol. We prove that the upper bound for any graph is $O(n^2)$ and we show that this bound is tight in a sense that there exists a graph for which the stopping time of algebraic gossip is $\Omega(n^2)$. Our general upper bound $O(\Delta n)$ is provided as a function of the maximum degree Δ of a graph and thus we can obtain a tight linear bound of $\Theta(n)$

for any graph with a constant maximum degree. Moreover, our results hold for $q \geq 2$ (coefficients field size), while previous results were for the case $q \geq n$.

In [8], we originally asked the following question. What are the properties of a network (beyond the maximum degree Δ) that capture the stopping time of algebraic gossip? To illustrate that the maximum degree is not always the correct metric, note the interesting observation that on the extended-barbell graph (Fig. 2.8) the stopping time of algebraic gossip is linear. So, by adding a single node to the barbell graph (Fig. 2.4 (b)) the stopping time has been changed by an order of magnitude?! A recent paper by Haeupler [33] makes a significant progress in answering our open question. In Section 3.1.2, we briefly describe this very interesting work and compare it to our results.

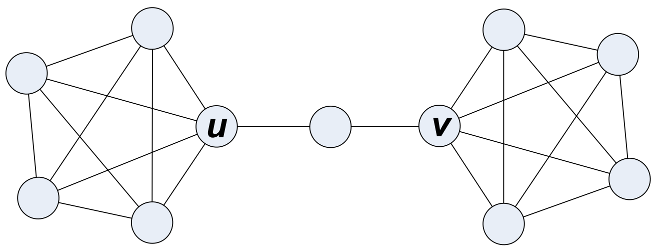


Figure 2.8: Extended barbell graph: additional node between the cliques.

In the next chapter (which describes our most recent work on algebraic gossip), we successfully address the topics of many-to-all communication and the non-uniform gossip approach. These topics were originally raised in the conference version [8] of the current work. We first provide an upper bound for the many-to-all scenario and show that the bound is tight for various topologies (in particular, for graphs with a constant maximum degree); second, we study a non-uniform gossip and propose a modified algebraic gossip algorithm that is order optimal for many families of graphs.

Chapter 3

Algebraic Gossip – k-Dissemination

3.1 Introduction

One of the most basic information spreading applications is that of disseminating information stored at a subset of source nodes to a set of sink nodes. Here we consider the k-*dissemination* case: k initial messages ($k \leq n$) located at some nodes (a node can hold more than one initial message) need to reach all n nodes. The *all-to-all communication* – each of n nodes has an initial value that is needed to be disseminated to all nodes – is a special case of k-dissemination. The goal is to perform this task in the lowest possible number of time steps when messages have *limited* size (i.e., a node may not be able to send all its data in one message).

Gossiping, or rumor-spreading, is a simple stochastic process for dissemination of information across a network. In a synchronous *round* of gossip, *each* node chooses a *single* neighbor as the *communication partner* and takes an action. In an asynchronous time model, at every timeslot, a single node wakes up and chooses a *communication partner*. Every n consecutive timeslots are considered as one *round*. The *gossip communication model* defines how to select this neighbor, e.g., *uniform* gossip is when the communication partner is selected uniformly at random from the set of all neighbors. We then consider three possible actions: either the node pushes information to the partner (`PUSH`), pulls information from the partner (`PULL`), or does both (`EXCHANGE`), but here we mostly present results about EXCHANGE.

A *gossip protocol* uses a gossip communication model in conjunction with the choice of the particular content that is exchanged. Due to their distributed nature, gossip protocols have gained popularity in recent years and have found applications both in communication networks (for example, updating database replicated at many sites [24, 39], computation of aggregate information [40] and multicast via network coding [22], to name a few) as well as in social networks [41, 17].

In this chapter we continue to analyze *algebraic gossip* which is a type of network coding known as random linear coding (RLNC) [45, 43] that uses gossip algorithms for all-to-all communication and k-dissemination. In algebraic gossip the content of messages is the random linear combination of all messages stored at a sender. Once a node has received enough independent messages (independent linear equations) it can solve the system of linear equations and discover all the initial values of all other nodes. It has been proved in [22] that using algebraic gossip can speedup message dissemination by an order of magnitude, compared to the uncoded dissemination scheme – "random message selection". In [36], authors showed that network coding can improve the throughput of the network by better sharing of the network resources. Note, however, that in gossip protocols, nodes select a single partner, so for k-dissemination to succeed each node needs to receive at least k messages (of bounded size), hence at least a total of kn messages need to be sent and received. This immediately leads to a trivial lower bound of $\Omega(k)$ rounds for k-dissemination.

We study uniform and non-uniform algebraic gossip both in the synchronous and the asynchronous time models on arbitrary graph topologies. The stopping time obviously depends on the protocol, the gossip communication model, the graph topology, but also on the time model, as shown in other cases [31].

3.1.1 Overview of Results of the Current Chapter

Our first set of results is about the stopping time of uniform algebraic gossip. In Chapter 2 we have shown a tight bound of $\Theta(n)$ for all-to-all communication for graphs with constant maximum degree. To prove this, we used a reduction of gossip to a network of queues and analyzed the waiting times in the queues. Bounding the general k-dissemination case is significantly harder, despite some similarity in the tools used. Unless explicitly stated, all our results are for gossip using `EXCHANGE` and are with high probability[1].

We provide a novel upper bound for uniform algebraic gossip of $O((k + \log n + D)\Delta)$ where D is the diameter and Δ is the maximum degree in the graph. For graphs with constant maximum degree ($\Delta = O(1)$) this leads to a bound of $O(k + D)$. In this case, we also show a matching lower bound of $\Omega(k + D)$ which makes uniform algebraic gossip an order **optimal** gossip protocol for these graphs.

However, there are topologies for which uniform algebraic gossip performs badly, e.g., in the barbell graph (two cliques connected with a single edge) it takes $\Omega(n^2)$ rounds to perform all-to-all communication (as we have shown in Chapter 2, Theorem 2.4). This is usually the result of bottlenecks that exist in the graph and lead to low conductance. For such "bad" topologies we propose here a modification of the uniform algebraic gossip called *Tree based Algebraic Gossip* (TAG). The basic idea of the protocol is that it operates in two phases: first, using a gossip protocol $\mathcal{S}$ it generates a spanning tree in which each node in the tree has a single parent. In the next phase, algebraic gossip is performed on the tree where each node does `EXCHANGE` with its parent. Let $t(\mathcal{S})$ and $d(\mathcal{S})$ be the stopping time of $\mathcal{S}$ and the diameter of the tree generated by $\mathcal{S}$, respectively. For any spanning tree gossip protocol $\mathcal{S}$ we prove for TAG an upper bound of: $O(k + \log n + d(\mathcal{S}) + t(\mathcal{S}))$ for the *synchronous* and the *asynchronous* time models. As a special case of a spanning tree protocol, one can use a

[1]An event occurs with high probability (*w.h.p.*) if its probability is at least $1 - O(\frac{1}{n})$.

<table>
<tr><th>Protocol</th><th>Graph</th><th>Synchronous</th><th>Asynchronous</th></tr>
<tr><td rowspan="2">Uniform AG</td><td>any graph</td><td colspan="2">$O((k + \log n + D)\Delta)$</td></tr>
<tr><td>constant max degree</td><td colspan="2">$\mathbf{\Theta(k + D)}$</td></tr>
<tr><td rowspan="3">TAG</td><td rowspan="2">any graph</td><td colspan="2">$O(k + \log n + d(\mathcal{S}) + t(\mathcal{S}))$</td></tr>
<tr><td>$O(k + \log n + t(\mathcal{B}))$</td><td>$O(k + \log n + d(\mathcal{B}) + t(\mathcal{B}))$</td></tr>
<tr><td>$k = \Theta(n)$, any graph</td><td colspan="2">$\mathbf{\Theta(n)}$</td></tr>
</table>

Table 3.1: Overview of the main results of Chapter 3. **Bold text** and Θ indicate order optimal result. D – diameter of the graph, Δ – maximum degree of the graph, $\mathcal{S}$ – spanning tree protocol, $\mathcal{B}$ – broadcast (or 1-dissemination) protocol, $d(\cdot)$ – diameter of the spanning tree generated by protocol $(\cdot)$, $t(\cdot)$ – stopping time of protocol $(\cdot)$.

gossip broadcast (or 1-dissemination) protocol $\mathcal{B}$ – a protocol in which a single message originated at some node should be disseminated to all nodes. Interestingly, using a gossip broadcast for the spanning tree construction in TAG, eliminates the dependence on the diameter of the spanning tree in the synchronous time model, i.e., if we use $\mathcal{B}$ as $\mathcal{S}$, we obtain the bound of $O(k + \log n + t(\mathcal{B}))$ rounds. For a general spanning tree protocol $\mathcal{S}$, it follows directly that if $k = \Omega(\max(\log n, d(\mathcal{S}), t(\mathcal{S})))$, TAG is an order **optimal** with a stopping time of $\Theta(k)$. We provide an example of this scenario which leads to the most significant result of the chapter. Using a simple round-robin-based broadcast we show that TAG is an order optimal gossip protocol for k-*dissemination* in any topology when $k = \Omega(n)$. This implies, somewhat surprisingly, that for **any graph**, if $k = \Theta(n)$, TAG finishes in $\Theta(n)$ rounds. In the barbell graph mentioned above, TAG leads to a speedup ratio of n compared to the uniform algebraic gossip. Table 3.1 summarizes our main results of the chapter and next, we discuss previous results.

3.1.2 Related Work

Uniform algebraic gossip was first proposed by Deb *et al.* in [22]. The authors studied uniform algebraic gossip using `PULL` and `PUSH` on the *complete graph* and showed a tight bound of $\Theta(k)$, for the case of $k = \omega(\log^3(n))$ messages. Boyd *et al.* [10, 12] studied the stopping time of a gossip protocol for the *averaging problem* using the `EXCHANGE` algorithm. They gave a bound for symmetric networks that is based on the second largest eigenvalue of the transition matrix or, equally, the mixing time of a random walk on the network, and showed that the mixing time captures the behavior of the protocol. Mosk-Aoyama and Shah [47] used a similar approach to [10, 12] to first analyze algebraic gossip on arbitrary networks. They consider symmetric stochastic matrices that (may) lead to a non-uniform gossip and gave an upper bound for the `PULL` algorithm that is based on a measure of conductance of the network. As the authors mentioned, the offered bound is not tight, which indicates that their conductance-based measure does not capture the full behavior of the protocol.

In [8], we used queuing theory as a novel approach for analyzing algebraic gossip. We then gave an upper bound of $O(n\Delta)$ rounds for any graph for the case of all-to-all communication, where Δ is the maximum degree in the graph. In addition, a lower bound of $\Omega(n^2)$ was obtained for the barbell graph – the worst case graph for algebraic gossip. The bounds (upper and lower) in [8] were tight in the sense that they matched each other for the worst case scenario. The parameter Δ is simple and convenient to use, but, it does not fully capture the behavior of algebraic gossip. While it gives optimal ($\Theta(n)$) result for any constant-degree graphs (e.g., line, grid), it fails to reflect the stopping time of algebraic gossip on the complete graph, for example, by giving the $O(n^2)$ bound instead of $O(n)$.

A recent work of Haeupler [33] is the most related to our work. Haeupler's paper makes a significant progress in analyzing the stopping time of algebraic gossip. While all previous works on algebraic gossip used the notion of *helpful message/node* to look at the rank evaluation of the matrices each node maintains (this approach was initially proposed by [22]), Haeupler used a completely different approach. Instead of looking on the growth of the node's subspace (spanned by the linear equations it has), he proposed to look at the orthogonal complement of the subspace and then analyze the process of its disappearing. This elegant and powerful approach led to very impressive results which apply also to adversarial dynamic networks and arbitrary edge probabilities. For the all-to-all communication scenario, a tight bound of $\Theta(\frac{n}{\gamma})$ was proposed, where γ is a min-cut measure of a related graph. This bound perfectly captures algebraic gossip behavior for any network topology. For the case of k-dissemination, the author gives a conjecture that the upper bound is of the form of $O(k+T)$ where T is the time to disseminate a single message to all the nodes. But formally, the bound that is proved is $O(k/\gamma+\log^2 n/\lambda)$ where λ is a conductance-based measure of the graph (Lemma 7.6 in [33]). The work in [33] implicitly considered the uniform algebraic gossip, but could be extended to non-uniform cases. It is therefore hard to compare TAG to the results of [33], nevertheless, our bounds for the uniform algebraic gossip are better for certain families of graphs. Table 3.2 presents few such examples.

Graph	$O(k/\gamma + \log^2 n/\lambda)/n$ [33]	$O((k + \log n + D)\Delta)$ [here]	Improvement factor
Line	$O(k + n\log^2 n)$	$O(k + n)$	$\log^2 n$
Grid	$O(k + \sqrt{n}\log^2 n)$	$O(k + \sqrt{n})$	$\log^2 n$ for $k = O(\sqrt{n})$
Binary Tree	$O(k + n\log^2 n)$	$O(k + \log n)$	$\Omega(\frac{n\log n}{k})$

Table 3.2: Comparison of our results with [33].

We would like to note that Haeupler [34] recently extended the results of [33] using the same techniques above to provide additional tighter bounds similar to the results we present here.

To give a quick summary of our results and previous work, the two main contributions of the chapter are i) we prove that for graphs with constant maximum degree uniform algebraic gossip is order optimal for k-dissemination in the synchronous time model and ii) we offer a new non-uniform algebraic gossip protocol, TAG, that is order optimal for large selections of graphs and k. The rest of the chapter is organized as follows: in Section 3.2 we give definitions. Section 3.3 proves results for uniform algebraic gossip and Section 3.4 presents the TAG protocol and its general bound. Section 3.4.2, then, discussses a case where TAG is optimal.

3.2 Preliminaries

As in Chapter 2, we model the communication network by a connected undirected graph $G_n = G_n(V, E)$, where V is the set of vertices and E is the set of edges. The number of vertices in the graph is $|V| = n$. Let $N(v) \subseteq V$ be a set of neighbors of node v and $d_v = |N(v)|$ its degree, let $\Delta = \max_v d_v$ be the maximum degree of G_n, and let D be the diameter of the graph.

We consider two time models: asynchronous and synchronous. In the *asynchronous* time model at every **timeslot**, **one node** selected independently and uniformly at random, takes an action and a single pair of nodes communicates. We consider n consecutive timeslots as one *round*. In the *synchronous* time model at every **round**, **every node** takes an action and selects a single communication partner. It is assumed that the information received in the current round will be available to a node for sending only from the beginning of the next round. A **Gossip communication model** (sometimes called gossip algorithm) defines the way information is spread in the network. In the gossip communication model, a node that wakes up (according to the time model) can initiate communication only with a single neighbor[2] (i.e., communication partner). The model describes how the communication partner is chosen and in which direction (to – `PUSH`, from –`PULL`, or both – `EXCHANGE`) the message is sent. In this chapter we use the following communication models:

Definition 3.1 (Uniform Gossip)**.** *Uniform gossip is a gossip in which a communication partner is chosen randomly and uniformly among all the neighbors.*

Definition 3.2 (Round-Robin ($\mathcal{RR}$) Gossip)**.** *In round-robin gossip, the communication partner is chosen according to a fixed, cyclic list, of the nodes' neighbors. This list dictates the order in which neighbors are being contacted.*

Notice that if the initial partner is chosen at random, the round-robin gossip communication model is known as the *quasirandom rumor spreading model* [1, 25].

3.2.1 Gossip Protocols

Gossip protocols define the task and the message content. In turn, a gossip protocol can use any of the gossip communication models defined above (and others). We will use two types of gossip protocols here: **Algebraic Gossip** and **Spanning Tree Gossip** protocols.

In this chapter, we define algebraic gossip as a k-dissemination protocol, i.e., its task is to deliver all the k messages, initially located at arbitrary nodes, to every node in the network. In algebraic gossip, every message is sent by a node according to the random linear coding (RLNC) technique which is described in Section 2.2.3. Here, we just remind the following definition which is necessary for understanding the concept of helpfulness in the analysis of algebraic gossip.

[2]Note that this implies that in the synchronous model a node can communicate with more than a single neighbor, if other nodes initiate communication with it.

Definition 3.3 (Helpful node ([22]) and helpful message)**.** *We say that a node x is a* **helpful node** *to a node y if and only if a random linear combination constructed by x can be linearly independent with all equations (messages) stored in y. We call a message a* **helpful message** *if it increases the dimension (or rank) of the node (i.e., the rank of the matrix in which the node stores the messages).*

A spanning tree gossip protocol, which we denote by $\mathcal{S}$, will create a *spanning tree* of a given graph, i.e., by its completion, every node, except the *root*, will have a single neighbor called the *parent*. Note that one simple way to generate a spanning tree is by using a 1-dissemination protocol, namely a broadcast protocol initiated by an arbitrary node that will disseminate its message (or ID) to every other node. Spanning tree gossip protocol will be used as an auxiliary protocol for the k-dissemination task along with the algebraic gossip (the resulting combined protocol we call TAG and formally describe it in Section 3.4). The list of notations, used throughout the chapter, can be found in Table 3.3.

3.3 k-dissemination with Uniform Algebraic Gossip

The main result of this section is that uniform algebraic gossip is order optimal k-dissemination for graphs with constant maximum degree and for any selection of k. It is formally stated in Theorem 3.3 and is an almost direct result of the following general bound for uniform algebraic gossip:

Theorem 3.1. *For any connected graph G_n, the stopping time of the uniform algebraic gossip protocol with k messages is $O((k + \log n + D)\Delta)$ rounds for synchronous and asynchronous time models* w.h.p.

n	Number of nodes
k	Number of messages needed to be disseminated
G_n	Connected graph with n nodes
T_n	Connected Tree graph with n nodes
$l_{\max}$	Depth of the tree created by a broadcast protocol
Q_n^{tree}	Network of n queues arranged in a tree topology
$Q_{l_{\max}}^{line}$	Network of $l_{\max}$ queues arranged in a line topology
D	Diameter of a graph
$N(v)$	Set of neighbors of the node v
d_v	Degree of the node v ($d_v = \|N(v)\|$)
Δ	Maximum degree of the graph ($\Delta = \max_v d_v$)
timeslot	Unit of time in the **asynchronous** time model
round	Unit of time in the **synchronous** time model (1 round = n timeslots)
$\mathcal{S}$	Some spanning tree gossip protocol
$\mathcal{B}$	Some broadcast (1-dissemination) gossip protocol
$d(\mathcal{S})$, $d(\mathcal{B})$	Diameter of the spanning tree created by the protocol
$\mathcal{RR}$	Round-robin communication model
$\mathcal{B}_{\mathcal{RR}}$	Broadcast gossip protocol based on the round-robin communication model
TAG	k-dissemination protocol that uses algebraic gossip and a spanning tree protocol
$t(\cdot)$	Stopping time of protocol $(\cdot)$
$t(Q_n^{tree})$	Stopping time of a queuing system – time by which all customers leave the system

Table 3.3: Notations used in the chapter.

The idea of the proof relies on the queuing networks technique we presented in Chapter 2. The major steps of the proof are:

- Consider a Breadth First Search (BFS) tree of G_n, T_n rooted at an arbitrary node v. The maximum depth ($l_{\max}$) of the tree is at most D (diameter of G_n).

- Reduce the problem of algebraic gossip on a tree T_n to a simple system of queues Q_n^{tree} rooted at v, where at each node we assume an infinite queue with a single server. Every initial message becomes a customer in the queuing system. The root v finishes once all the customers arrive at it.

- Show that the stopping time of the tree topology queuing system – Q_n^{tree}, is $O((k+\log n+l_{\max})n\Delta)$ timeslots *w.h.p.* So, we obtain the stopping time for the node v.

- Use union bound to obtain the result for all the nodes in G_n.

Just before we start the formal proof of Theorem 3.1, we present an interesting theorem related to queuing theory. The theorem gives the stopping time of the feedforward queuing system [20] arranged in a tree topology. In the feedforward network, a customer can not enter the same queue more than once , thus, customers are always forwarded towards the root and eventually leave the system via the queue at the root of the tree. Consider the following scenario: n identical M/M/1 queues (M/M/1 system is a queue with a single server in which interarrival and service times are distributed exponentially) arranged in a tree topology. There are no external arrivals, and there are k customers arbitrarily distributed in the system. We ask the following question: how much time will it take for the last customer to leave the system?

Theorem 3.2. *Let Q_n^{tree} be a network of n nodes arranged in a tree topology, rooted at the node v. The depth of the tree is $l_{\max}$. Each node has an infinite queue, and a single exponential server with parameter μ. The total amount of customers in the system is k and they are initially distributed arbitrarily in the network. The time by which all the customers leave the network via the root node v is $t(Q_n^{tree}) = O((k + l_{\max} + \log n)/\mu)$ timeslots with probability of at least $1 - \frac{2}{n^2}$.*

Proof. The main idea of the proof is to show that the stopping time of the network Q_n^{tree} (i.e., the time by which all the customers leave the network) is stochastically smaller or equal (see Definition 2.3 below) to the stopping time of the systems of $l_{\max}$ queues arranged in a line topology – $Q^{line}_{l_{\max}}$. Then, we make the system $Q^{line}_{l_{\max}}$ stochastically slower by moving all the customers out of the system and make them enter back via the farthest queue with the rate $\lambda = \mu/2$. Finally, we use Jackson's Theorem for open networks to find the stopping time of the system. See Fig. 2.3 for the illustration. The formal proof of the theorem is analogous to the proof of Theorem 2.3 with the only change that the total number of customers is $k \leq n$ instead of n. So, following the proof of Theorem 2.3 we can conclude that:

$$t(Q^{tree}_n) \preceq t(\hat{Q}^{tree}_n) \approx t(Q^{line}_{l_{\max}}) \preceq t(\grave{Q}^{line}_{l_{\max}}) \preceq t(\hat{Q}^{line}_{l_{\max}}). \tag{3.1}$$

Now, it is left to show that $t(\hat{Q}^{line}_{l_{\max}}) = O((k + l_{\max} + \log n)/\mu)$ with high probability.

Lemma 3.1. *The time it will take to the last customer to leave the system $\hat{Q}^{line}_{l_{\max}}$ ($l_{\max}$ M/M/1 queues arranged in a line topology) is $O((k + \log n + l_{\max})/\mu)$ with probability of at least $1 - \frac{1}{n^2}$.*

Proof. Initially, all the customers (from now we will call them *real* customers) are located in the last ($Z^{l_{\max}}_1$) queue. We now take all the *real* customers out of this queue and will make them enter the system (via the $Z^{l_{\max}}_1$) from outside. We define the *real* customers' arrivals as a Poisson process with rate $\lambda = \frac{\mu}{2}$. So, $\rho = \frac{\lambda}{\mu} = \frac{1}{2} < 1$ for all the queues in the system. Clearly, such an assumption only increases the stopping time of the system (stopping time is the time until the last customer leaves the system).

According to Jackson's theorem (Section 2.3), there exists an equilibrium state. So, we need to ensure that the lengths of all queues at time $t = 0$ are according to the equilibrium state probability distribution. We add *dummy* customers to all the queues according to the stationary distribution. By adding additional *dummy* customers to the system, we make the *real* customers wait longer in the queues, thus increasing the stopping time.

We will compute the stopping time $t(\hat{Q}^{line}_{l_{\max}})$ in two phases: Let us denote this time as $t_1 + t_2$, where t_1 is the time needed for the k'th customer to arrive at the first queue, and t_2 is the time needed for the k'th customer to pass through all the $l_{\max}$ queues in the system.

From Jackson's theorem, it follows that the number of customers in each queue is independent, which implies that the random variables that represent the waiting times in each queue are independent.

The random variable t_1 is the sum of k independent random variables distributed exponentially with parameter $\mu/2$. From Lemma 2.3 we obtain that t_2 is the sum of $l_{\max}$ independent random variables distributed exponentially with parameter $\mu - \lambda = \mu/2$. $\mathrm{E}\left[t_1\right] = \sum_{i=1}^{k} 2/\mu = 2k/\mu$, and by taking $\alpha = 2 + 4\frac{\ln n}{k}$, we obtain:

$$\begin{aligned} \Pr\left(t_1 < (4k + 8\ln n)/\mu\right) &> 1 - (2e^{-(2+4\frac{\ln n}{k})/2})^k && (3.2)\\ &= 1 - (\tfrac{2}{e})^k e^{-2\ln n} && (3.3)\\ &\geq 1 - e^{-2\ln n} && (3.4)\\ &\geq 1 - \tfrac{1}{n^2}. && (3.5) \end{aligned}$$

In a similar way we obtain:

$$\Pr\left(t_2 < (4l_{max} + 8\ln n)/\mu\right) > 1 - \tfrac{1}{n^2}. \qquad (3.6)$$

$t(\hat{Q}^{line}_{l_{\max}}) = t_1 + t_2$, thus, using union bound:

$$\Pr\left(t_1 + t_2 < (4k + 4l_{max} + 16\ln n)/\mu\right) > 1 - \tfrac{2}{n^2} \qquad (3.7)$$

and thus:

$$t(\hat{Q}^{line}_{l_{\max}}) = O((k + l_{\max} + \log n)/\mu) \qquad (3.8)$$

w.p. of at least $1 - \frac{2}{n^2}$.

□

We now complete the proof of Theorem 3.2. Since $t(Q_n^{tree}) \preceq t(\hat{Q}_{l_{\max}}^{line})$ and thus, using Lemma 3.1: $t(Q_n^{tree}) = O((k + l_{\max} + \log n)/\mu)$ w.p. of at least $1 - \frac{2}{n^2}$.

□

We can now prove Theorem 3.1.

Proof of Theorem 3.1. We start the analysis of the uniform algebraic gossip with k messages and the asynchronous time model. First, we consider a Breadth First Search (BFS) spanning tree T_n of G_n rooted at an arbitrary node v. The depth of T_n is $l_{\max}$, and since T_n is the shortest path tree, $l_{\max} \leq D$, where D is the diameter of the graph. On the tree T_n, consider a message flow towards the root v from all other nodes. Once k *helpful messages* arrive at v, it will reach rank k and finish the algebraic gossip protocol. We ignore messages that are not sent in the direction of v. Ignoring part of messages can only increase the stopping time of the algebraic gossip protocol.

We define a queuing system Q_n^{tree} by assuming an infinite queue with a single server at each node. The root of Q_n^{tree} is the node v. Customers of our queuing network are *helpful messages*, i.e., messages that increase the rank of a node they arrive at. This means that every customer arriving at some node increases its rank by 1. When a customer leaves a node, it arrives at the parent node. The queue length of a node represents a measure of *helpfulness* of the node to its parent, i.e., the number of *helpful messages* it can generate for it.

The service procedure at a node is a transmission of a *helpful message* towards the node v (from a node to its parent). Lemma 2.1 in [22] gives a lower bound for the probability of a message sent by a *helpful node* to be a *helpful message*, which is: $1-\frac{1}{q}$, where q is a size of a finite field $\mathbb{F}_q$ from which the random network coding coefficients are drawn. In the uniform gossip communication model, the communication partner of a node is chosen randomly among all the node's neighbors in the original graph G_n. The degree of each node in G_n is at most Δ. Thus, in the asynchronous time model, in a given timeslot, a *helpful message* will be sent over the edge in a specific direction with probability of at least $(1 - \frac{1}{q})/n\Delta$, where $\frac{1}{n}$ is the probability that a given node wakes up in a given timeslot, $\frac{1}{\Delta}$ is the minimal probability that a specific partner (the parent of the node) will be chosen, and $1 - \frac{1}{q}$ is the minimal probability that the message will be *helpful*. Thus, we can consider that the service time in our queuing system is geometrically distributed with parameter $p \geq (1 - \frac{1}{q})/n\Delta$, and since $q \geq 2$, we can assume the worst case: $p = \frac{1}{2n\Delta}$.

Lemma 2.2 shows that we can model the service time of each server as an exponential random variable with parameter $\mu = p$, since in this case, exponential servers are stochastically *slower* than geometric. Such an assumption can only increase the stopping time.

Theorem 3.2 with $\mu = p$ gives us an upper bound for the stopping time of the node v, $t_v = O((k + l_{\max} + \log n)2n\Delta)$ timeslots with probability of at least $1 - \frac{2}{n^2}$. Since the depth of every BFS tree is bounded by the diameter D, using a union bound we obtain the upper bound (in timeslots) for all the nodes in G_n:

$$\Pr\left(\bigcap_{v \in V} t_v = O((k + \log n + D)2n\Delta)\right) > 1 - \frac{2}{n}. \tag{3.9}$$

Thus we obtain the upper bound for uniform algebraic gossip: $O((k + \log n + D)\Delta)$ rounds. Next, we show that this bound holds also for the synchronous time model. The proof for the synchronous time model is almost the same as in the asynchronous case, except for the following change. Instead of dividing time into timeslots, we measure it by rounds (1 round = n timeslots). In a given **round**, a *helpful message* will be sent over the edge in a specific direction with probability $p \geq (1 - \frac{1}{q})/\Delta$, where the $\frac{1}{\Delta}$ is the minimal probability that a specific partner (the parent of the node) will be chosen, and $1 - \frac{1}{q}$ is the minimal probability that the message will be *helpful*. Since $q \geq 2$, we can assume the worst case: $p = \frac{1}{2\Delta}$. The difference from the asynchronous model is the factor of n in p, since in the synchronous model, every node wakes up exactly once in a each round. Moreover, in the synchronous case (and in the `EXCHANGE` gossip variation) there is a possibility to receive 2 messages from the same node in one round (in the asynchronous time model it was impossible to receive 2 messages from the same node in one timeslot). We assume that if a node receives 2 messages from the same node at the same round, it will discard the second one. Such an assumption can only increase the stopping time of the protocol, and will make our analysis simpler. From that point on, the analysis is analogous to the asynchronous case since Theorem 3.2 does not depend on the time model. □

3.3.1 Optimality for Constant Maximum Degree Graphs

Following Theorem 3.1 we can state the main results of the section:

Theorem 3.3. *For any connected graph G_n with constant maximum degree, the stopping time of the uniform algebraic gossip protocol with k messages is $\Theta(k + D)$* w.h.p. *in the synchronous and asynchronous time models.*

Proof. To show the upper bound, we use the following simple claim:

Claim 3.1. *For any connected graph G_n with maximum degree Δ and diameter D:*

$$D \geq \log_{\Delta} n - 1.$$

Proof. Let us sum up all the n vertices of G_n in the following way. We start with an arbitrary node v and count it as 1. Then we split the sum of n vertices into D parts, where D is the diameter of G_n. Each part represents number of vertices located at the distance i ($i \in [0, .., D]$) from the node v. Since we are interested in the lower bound on D, we can assume the maximum degree for every node (so, the number of parts in the sum will be minimal). We define n_i ($i \in [0, .., D]$) as the number of vertices located at the distance i from the node v. Thus we obtain:

$$n_0 + n_1 + n_2 + \cdots + n_D = n \tag{3.10}$$

$$1 + \Delta + \Delta^2 + \cdots + \Delta^D \geq n \tag{3.11}$$

$$\frac{\Delta^{D+1} - 1}{\Delta - 1} \geq n \tag{3.12}$$

$$\Delta^{D+1} \geq n \tag{3.13}$$

$$D \geq \log_\Delta n - 1. \tag{3.14}$$

□

Now, using Claim 3.1 and the fact that the maximum degree is constant (i.e., $\Delta = O(1)$ and thus: $D = \Omega(\log n)$), the upper bound follows. For the lower bound, note that in order to disseminate k messages to n nodes, at least kn transmissions should occur in the network. In synchronous time model, kn transmissions require at least $k/2$ rounds, since every round at most $2n$ messages are sent (2 transmissions per communication pair). In the asynchronous time model, kn transmissions require at least $kn/2$ timeslots, since at each timeslot at most 2 nodes transmit (due to `EXCHANGE`). Thus, in both time models, $\Omega(k)$ rounds are required. Moreover, in the synchronous time model, dissemination of a single message will take at least D rounds, since in this model, a message can travel at most one hop in a single round. So, for the synchronous time model, the bound $\Theta(k+D)$ is tight and optimal. The last thing we have to show is that for the asynchronous time model, with high probability we will need at least $\Omega(nD)$ timeslots which are $\Omega(D)$ rounds.

Consider two nodes u and v with distance D between them. We will show that, with high probability, a message will not travel for a distance D (or larger) from u in less than $\frac{Dn}{2\Delta^3}$ timeslots. Thus, it is impossible to finish the algebraic gossip protocol in less than $\frac{Dn}{2\Delta^3}$ timeslots. Let $X = \sum_{i=1}^{D} X_i$, where $X_i \sim \text{Geom}(2/n)$, be the number of timeslots needed to cross a path of D edges. Notice, that $2/n$ is a maximum probability of sending a helpful message on a given edge in the `EXCHANGE` communication model. From Claim 3.1 we have that $D \geq \log_{\Delta} n - 1 = \frac{\log_2 n}{\log_2 \Delta} - 1 \geq \frac{\log_2 n}{2 \log_2 \Delta}$.

We will use now Lemma 2.6 with: $m = D$, $p = \frac{2}{n}$, $k = \frac{1}{\Delta^3} \frac{m}{p} = \frac{Dn}{2\Delta^3}$ and will obtain:

$$\Pr\left(X \leq \frac{Dn}{2\Delta^3}\right) \leq \left(e^{(-1+\Delta^{-3})}\Delta^{-3}\right)^D \tag{3.15}$$

$$\leq \Delta^{-3D}. \tag{3.16}$$

Since there are at most Δ^D possible paths of length D starting at u, we can use union bound to obtain the probability that the number of timeslots needed to travel to a distance D is at most $\frac{Dn}{2\Delta^3}$. This probability will be at most: $\Delta^{-3D}\Delta^D = \Delta^{-2D}$. By taking the smallest value of D, we get the worst case probability: $\Delta^{-2D} = \Delta^{-2\frac{\log_2 n}{2\log_2 \Delta}} = 1/n$. Thus, we obtain that for $\Delta = O(1)$, stopping time of algebraic gossip is $\Omega(Dn)$ timeslots with high probability. So, also for the asynchronous time model, the bound $\Theta(k + D)$ is tight and optimal. □

3.4 TAG: k-dissemination with Tree-based Algebraic Gossip

We now describe the protocol TAG (Tree based Algebraic Gossip), which is a k-dissemination gossip protocol that exploits algebraic gossip in conjunction with a spanning tree gossip protocol $\mathcal{S}$ (see Sec. 3.2). Given a connected network of n nodes and k messages $x_1, ..., x_k$ that are initially located at some nodes, the goal of the protocol TAG is to disseminate all the k messages to all the n nodes. The protocol consists of two phases. Both phases are performed simultaneously in the following way: if a node wakes up[3] and the total number of its wakeups until now is even (we

[3] wakes up – selected according to the time model for a communication action.

Protocol TAG Pseudo code for node v. Example for asynchronous time model.

Require: $N(v)$, k, gossip spanning tree protocol $\mathcal{S}$

Initialize: $parent = null$ // the *parent* will be set up by $\mathcal{S}$ according to the received messages.

On odd wakeup: // Phase 1: gossip spanning tree protocol $\mathcal{S}$

choose parter $u \in N(v)$ and exchange messages with it according to $\mathcal{S}$

On even wakeup: // Phase 2: algebraic gossip

if obtained *parent* during the protocol $\mathcal{S}$ **then**

    exchange messages with *parent* according to algebraic gossip (RLNC)

On contact from other node $w \in N(v)$:

if w performs Phase 1 **then**

    exchange messages with w according to $\mathcal{S}$

else(w performs Phase 2)

    exchange messages with w according to algebraic gossip (RLNC)

call such a wakeup an *even wakeup*), it acts according to *Phase 1* of the protocol. If the node wakes up and the total number of its wakeups until now is odd (we call such a wakeup an *odd wakeup*), it acts according to *Phase 2* of the protocol.

- In Phase 1, a node performs a spanning tree gossip protocol $\mathcal{S}$. Once a node becomes a part of the spanning tree, it obtains a **parent**.
- In Phase 2, a node is idle until it obtains a **parent** in Phase 1. From now on, the node will perform an `EXCHANGE` algebraic gossip protocol with a fixed communication partner – its **parent**. Notice that the *root* node will never obtain a **parent**, but due to the `EXCHANGE` scheme, messages will be pushed to it and pulled from it by its children nodes.

The following theorem gives an upper bound on the stopping time of the protocol TAG.

Theorem 3.4. *Let $t(\mathcal{S})$ be the stopping time of the gossip spanning tree protocol $\mathcal{S}$ performed at Phase 1, and let $d(\mathcal{S})$ be the diameter of the spanning tree created by $\mathcal{S}$. For any connected graph G_n, the stopping time of the k-dissemination protocol TAG:*

$$t(TAG) = O(k + \log n + d(\mathcal{S}) + t(\mathcal{S})) \text{ rounds} \tag{3.17}$$

for synchronous and asynchronous time models, and w.h.p.

In order to prove this theorem, we will find the time needed to finish TAG, after Phase 1 is completed. Once Phase 1 is completed, every node knows its parent and thus, in Phase 2, we have the algebraic gossip `EXCHANGE` protocol on the spanning tree T_n, where communication partners of the nodes are their parents. The following lemma gives an upper bound on the stopping time of such a setting.

Lemma 3.2. *Let T_n be a tree with n nodes, rooted at the node r, with depth $l_{\max}$. There are k initial messages located at some nodes in the tree. Consider algebraic gossip `EXCHANGE` protocol with the following communication model: the communication partner of a node is fixed to be its parent in T_n during the whole protocol. Then, the time needed for **all the nodes** to learn all the k messages is $O(k+\log n+l_{\max})$ rounds for the synchronous and asynchronous time models, with probability of at least $1 - \frac{2}{n}$.*

The proof of Lemma 3.2 is very similar to the proof of Theorem 3.1, and relies on reducing the problem of algebraic gossip to a simple system of queues. The service time is geometrically distributed with a worst-case parameter $p = \frac{1}{2n}$. The Δ is eliminated from p since each node chooses now a single communication partner. Then, using Theorem 3.2 we obtain the stopping time of algebraic gossip with on the tree T_n. Following is the detailed proof of the lemma.

Proof. On T_n, consider a message flow towards an arbitrary node v (*not necessary the root* of T_n) from all other nodes. Once k *helpful messages* arrive at v, it will reach the rank k and finish the algebraic gossip protocol. Due to the proposed communication model, every node in T_n has a fixed communication partner – its parent, so, each edge e in the tree has at least one node which will issue, on its wakeup, a bidirectional communication (`EXCHANGE`) over e. Thus, from every node, a message can be sent towards v. We ignore messages that are not sent in the direction of v. Ignoring part of messages can only increase the stopping time of the algebraic gossip protocol.

As in the proof of Theorem 3.1, we define a queuing system Q_n^{tree} by assuming an infinite queue with a single server at each node. The root of Q_n^{tree} will be an arbitrary node v, and let $l_{\max}^v$ be the depth of the tree Q_n^{tree}. In our communication model, the communication partner of a node *is always its parent* in the tree. Thus, in the `EXCHANGE` gossip variation, in the asynchronous time model, in a given timeslot, a *helpful message* will be sent over the edge in a specific direction with probability of at least $(1 - \frac{1}{q})/n$. Thus, we can consider that the service time in our queuing system is geometrically distributed with parameter $p \geq (1 - \frac{1}{q})/n$, and since $q \geq 2$, we can assume the worst case: $p = \frac{1}{2n}$.

Using Theorem 3.2 for the tree T_n rooted at v, with $\mu = p$, we get an upper bound for the stopping time of the node v, $t_v = O((k + l_{\max}^v + \log n)2n)$ timeslots with probability of at least $1 - \frac{2}{n^2}$, where the $l_{\max}^v$ is the depth of the tree T_n rooted at v. Since $l_{\max}^v \leq 2l_{\max}$ (where $l_{\max}$ is the depth of T_n rooted at r), we can replace the $l_{\max}^v$ with $2l_{\max}$. So, using union bound, we obtain the upper bound (measured in timeslots) for all the nodes in T_n:

$$\Pr\left(\bigcap_{v \in V} t_v = O((k + \log n + l_{\max})2n)\right) > 1 - \frac{2}{n}. \tag{3.18}$$

As in the proof of Theorem 3.1, in the synchronous time model, the service time distribution parameter p will be larger by a factor of n, and the time will be measured in rounds instead of timeslots. Thus, using the same arguments as in the proof of Theorem 3.1, we obtain the upper bound of $O(k + \log n + l_{\max})$ rounds for the synchronous time model. Thus, the lemma holds for both time models. □

Proof of Theorem 3.4. Since for every choice of the tree root, the depth of the tree T_n (which was created using protocol $t(\mathcal{S})$) is bounded by its diameter, we can replace the $l_{\max}$ in the bound $O(k + \log n + l_{\max}))$ with $d(\mathcal{S})$. Now, we just add the stopping time of Phase 1 (the spanning tree time – $t(\mathcal{S})$) and the stopping time of Phase 2 (after Phase 1 has finished), and obtain that the number of rounds needed to complete the protocol TAG is $O(k + \log n + d(\mathcal{S}) + t(\mathcal{S}))$ *w.h.p.* □

3.4.1 1-dissemination as a Spanning Tree Protocol in TAG

The spanning tree task can be successfully performed by a simple gossip broadcast (or 1-dissemination) protocol. When a node receives for the first time the message, it marks the sending node as its parent. If more than one message was received during a single round, then an arbitrary message is selected and its sender is marked as a parent. In such a way we obtain a spanning tree rooted at the node that initiated the broadcast protocol. Let us denote a gossip 1-dissemination protocol as $\mathcal{B}$. Then, the result of Theorem 3.4 can be rewritten as: $t(\text{TAG}) = O(k + \log n + d(\mathcal{B}) + t(\mathcal{B}))$. An interesting observation regarding the broadcast protocol $\mathcal{B}$, is that for synchronous time model the depth of the broadcast tree cannot be larger that the broadcast time (measured in rounds), i.e., $t(\mathcal{B}) \geq d(\mathcal{B})$. The last is true since a message can not travel more than one hop in a single round. Thus, for the synchronous time model we obtain that the number of rounds needed to complete the TAG protocol *w.h.p.* is: $t(\text{TAG}) = O(k+\log n+t(\mathcal{B}))$. We summarize the above idea in the following corollary:

Corollary 3.1. *Let $\mathcal{B}$ be a a gossip 1-dissemination protocol. Then, the stopping time of the k-dissemination protocol TAG, is $t(TAG) = O(k + \log n + d(\mathcal{B}) + t(\mathcal{B}))$ for asynchronous time model, and $t(TAG) = O(k+\log n+t(\mathcal{B}))$ for the synchronous time model.*

3.4.2 Optimal All-to-all Dissemination Using TAG

In this section we propose to use the TAG protocol in conjunction with a 1-dissemination (or broadcast) gossip protocol $\mathcal{B}_{\mathcal{RR}}$ for spanning tree construction. For the case where $k = \Theta(n)$ messages need to be disseminated, TAG with $\mathcal{B}_{\mathcal{RR}}$ achieves order optimal performance. For the case $k = \Omega(n)$ the lower bound of any gossip dissemination protocol is $\Omega(n)$ rounds. The bound from Theorem 3.4 gives $t(\text{TAG}) = O(k + \log n + d(\mathcal{S}) + t(\mathcal{S}))$, and if $k = n$ we obtain $O(n + t(\mathcal{S}))$. Thus, all we need to show is the existence of a gossip spanning tree protocol that finishes after $O(n)$ rounds *w.h.p.* on any graph.

Theorem 3.5. *For any connected graph G_n, the stopping time of the broadcast protocol with the round-robin communication model – $\mathcal{B}_{\mathcal{RR}}$ is $O(n)$ rounds. In the asynchronous time model, this result holds with probability of at least $1 - n(2/e)^{3n}$, and in the synchronous time model, with probability 1.*

In order to prove Theorem 3.5 we need the following two lemmas. The first lemma gives an upper bound on the sum of degrees along any shortest path, and was presented in [26] (inside the proof of Theorem 2.1). For completeness, we give the proof of this lemma.

Lemma 3.3 ([26])**.** *For any connected graph G_n with n nodes, the sum of the degrees of the nodes along any shortest path between any two nodes v and u is at most $3n$.*

Proof. Without loss of generality, consider a BFS spanning tree of G rooted at some node v, and some arbitrary leaf u. We will find the maximum degree of the node located on the path $(v \to u)$ at distance i from the root v. Clearly, such a node can be connected only to the following nodes:

- Nodes that are located at distance $i-1$ from the root. (It can not be connected to the nodes that are closer to the root (than $i-1$) since then, its distance from the root would be $i-1$ which contradicts the given BFS execution.)

- Nodes that are at the same distance i from the root.
- Nodes that are located at distance $i+1$ from the root. (It can not be connected to the nodes that are farther from the root (than $i+1$) since then, their distance from the root would be $i+1$ which contradicts the given BFS execution.)

Let us define m_i as the number of nodes at distance i from the root. Clearly, $\sum_{i=0}^{n-1} m_i = n$. (The node at distance 0 is the root v). The degree of a node (at distance i from the root) can be at most: $d_i \leq (m_{i-1} + m_i + m_{i+1})$. Thus, the sum of degrees on a path of length l from the root to a leaf is at most: $d = \sum_{i=0}^{l} d_i$. Since $l \leq n-1$, $d = \sum_{i=0}^{l} d_i \leq \sum_{i=0}^{n-1} d_i = \sum_{i=0}^{n-1}(m_{i-1} + m_i + m_{i+1}) \leq 3n$. □

The second lemma gives an upper bound on the sum of m i.i.d. geometric random variables.

Lemma 3.4. *Let X be a sum of m independent and identically distributed geometric random variables (each one with parameter $p > 0$) and $E[X] = \frac{m}{p}$. Then, for $\alpha > 1$:*

$$\Pr\left(X \leq \alpha E[X]\right) > 1 - \left(\alpha e^{1-\alpha}\right)^m. \tag{3.19}$$

Proof. First, we will define Y as the sum of k independent Bernoulli random variables, i.e., $Y = \sum_{i=1}^{k} Y_i$, where $Y_i \sim Bernoulli(p)$. Let us notice that:

$$\Pr\left(X \leq k\right) = \Pr\left(Y \geq m\right) \tag{3.20}$$

The last is true since the event of observing at least m successes in a sequence of k Bernoulli trials implies that the sum of m independent geometric random variables is no more than k. On the other hand, if the sum of m independent geometric random variables is no more than k it implies that m successes occurred no later than the k-th trial and thus $Y \geq m$.

Now we will use a Chernoff bound for the sum of independent Bernoulli random variables presented in [46]: For any $0 < \delta < 1$ and $\mu = \mathrm{E}\left[Y\right]$:

$$\Pr\left(Y \leq (1-\delta)\mu\right) \leq \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu}. \tag{3.21}$$

Since $\mu = \mathrm{E}[Y] = kp$, and by letting $\delta = \frac{kp-m}{kp}$ we obtain:

$$\Pr(Y \leq (1-\delta)\mu) = \Pr(Y \leq m) \leq \left(\frac{m}{e^{\frac{m-kp}{m}}kp}\right)^{-m}. \tag{3.22}$$

$$\Pr(Y \geq m) > 1 - \left(\frac{m}{e^{\frac{m-kp}{m}}kp}\right)^{-m} \tag{3.23}$$

$$\Pr(X \leq k) > 1 - \left(\frac{m}{e^{\frac{m-kp}{m}}kp}\right)^{-m} \tag{3.24}$$

By substituting $k = \alpha\frac{m}{p} = \alpha\mathrm{E}[X]$ (where $\alpha > 1$) we obtain:

$$\Pr(X \leq \alpha\mathrm{E}[X]) > 1 - \left(\frac{e^{\alpha}}{e\alpha}\right)^{-m}. \tag{3.25}$$

□

Now we can prove the theorem.

Proof of Theorem 3.5. In this proof we assume the `PUSH` gossip variation, but it is clear that the result holds also for `EXCHANGE`.

Without loss of generality, assume that the message that needs to be disseminated is initially located at the node v. In the *round-robin* gossip, when a node is scheduled to transmit, it transmits a message to its neighbor according to the *round robin* scheme. I.e, at every transmission a message is sent to a different neighbor.

Consider a shortest path between v and some other node u. On the shortest path of length l there is exactly one node at the distance i from v, where $i \in [0,\ldots,l]$, and $l \leq n-1$. Let d_i be the degree of a node at the distance i from v. In order to guarantee the delivery of the message from v to u, we need $\sum_{i=0}^{l} d_i$ transmissions in the following order: first, we need d_0 transmissions of the node v, then d_1 transmissions of the next node in the path $v \to u$, and so on until the message is delivered to u. From Lemma 3.3, $\sum_{i=0}^{l} d_i \leq 3n$.

In the asynchronous model, a node transmits at a given *timeslot* with probability $\frac{1}{n}$. So, the number of timeslots until some specific node transmits is a geometric random variable with parameter $\frac{1}{n}$. We define this geometric random variable as X, i.e., $X \sim \text{Geom}\left(\frac{1}{n}\right)$. The number of timeslots until $3n$ specific transmissions occur, is the sum of $3n$ independent geometric random variables. Using Lemma 3.4 (with $\alpha = 2$) we obtain the bound of $O(n^2)$ timeslots (or $O(n)$ rounds) with exponential high probability. The last allows us to perform union bound for shortest paths to all other nodes in G, thus obtaining the $O(n)$ bound for the broadcast time.

It is easy to see that in the synchronous time model, $3n$ specific transmissions will occur exactly after $3n$ communication rounds. E.g., after d_0 rounds, v will perform d_0 transmissions – each one to different neighbor (according to the round-robin scheme). Thus, the message is delivered to u after at most $3n$ rounds with probability 1. □

Corollary 3.2. *Let $\mathcal{B}_{\mathcal{RR}}$ be a gossip* 1*-dissemination (broadcast) protocol with the round-robin communication model. Then, if $k = \Theta(n)$ and $\mathcal{S} = \mathcal{B}_{\mathcal{RR}}$, the stopping time of the k-dissemination protocol TAG, is $t(TAG) = \Theta(n)$.*

Proof. Using Theorems 3.4 and 3.5 we obtain the upper bound on the stopping time of TAG with $\mathcal{B}_{\mathcal{RR}}$ as a spanning tree construction protocol: $O(k + \log n + d(\mathcal{S}) + n)$ which is $\Theta(n)$ for $k = \Theta(n)$ and $\mathcal{S} = \mathcal{B}_{\mathcal{RR}}$. □

3.4.3 Tree Based Protocol: Discussion

The main contribution of the TAG protocol is not in proposing a practical dissemination approach but in making an additional step towards understanding the behavior of algebraic gossip. TAG is a tree based protocol and therefore it is natural to question the fitness of such protocol for gossiping and in particular network coding. A tree topology is obviously very vulnerable to edges failures, even if one of the tree edges disconnects, information dissemination will fail. But this could be solved with a more robust topology than a tree, a topology with several *outgoing edges* and not only one. This observation is a major contribution of the chapter, namely the distinction between outgoing and incoming edges in gossip protocols that are based on `EXCHANGE`. In

TAG, there is a single outgoing edge for each node, the edge that points to your parent, while the incoming degree is unbounded. In the gossip process, nodes only initiate `EXCHANGE` on their outgoing edges (uniformly at random). So for any topology $\mathcal{T}$, such that the maximum outgoing degree Δ_{out} is constant (e.g., in a spanning tree $\Delta_{out} = 1$), all our results for TAG hold. There are several ways to generate such robust topologies, one is to build several spanning trees instead of just one. In phase 1 we can build a constant number of spanning trees (let's say c trees) and thus a node in phase 2 will choose a specific parent with probability of $1/c$. Clearly, having a constant number of neighbors during the phase 2 will not change the asymptotic upper bound but will add a factor of robustness to the TAG protocol. More recently [16] proposed a mechanism that builds a sub-graph with diameter $O(D + \text{polylog}(n))$ (where D is the diameter of the original graph) and $\Delta_{out} = O(1)$ in time $O(\text{polylog}(n))$, using this sub-graph for algebraic gossip will give optimal results when $k > t$ where $t = O(\text{polylog}(n))$ is the running time. More generally, our results shift the focus of the problem of the stopping time of algebraic gossip to the problem of a fast generation of a sub-graph with bounded out degree on which the gossip will take place. Considering this, a major open problem is how to generate (via gossip) a topology $\mathcal{T}$ which is a sub-graph of the original graph, with diameter $O(D + t)$ and $\Delta_{out} = O(1)$ in time $O(t)$ where t is as small as possible. Our results indicate that algebraic gossip on $\mathcal{T}$ will be order optimal for any k. A related interesting question is about lower bounds for the running time.

Another important question about TAG is the need of coding messages when $\Delta_{out} = O(1)$ and in particular when $\Delta_{out} = 1$. If the topology for gossip is a tree, why can't we use a standard broadcast techniques without coding (i.e., mixing) messages? The question about the necessity of network coding was already raised before; in [27], the authors give a protocol for disseminating k messages in a complete graph in $O(k + \ln n)$ (which is optimal) without network coding but with an additional information exchange before the actual message transfer. The idea is that nodes asked their neighbors only for missing messages, such that every message sent is *helpful.* This can be done in our case as well, or in general in every gossip scheme. But when there is no bound on the incoming degree (as in our case) such a procedure will have to maintain checklists and request different messages from different neighbors. Network coding and algebraic gossip give a much simpler procedure that still guarantees with (enough) high probability sending/receiving *helpful* messages

3.5 Conclusions

In this chapter we have studied the problem of disseminating information from a subset of k nodes to all the n nodes on connected graphs. While Chapter 2 has been focused on the all-to-all dissemination problem (i.e., n-dissemination), the current chapter deals with k-disemination. We prove bounds for the uniform algebraic gossip which are optimal for some graph families (e.g., for graphs with a constant maximum degree). For some topologies, our bounds are better than any previously known results. Moreover, we propose here an alternative dissemination technique based on algebraic gossip (the TAG protocol) which is an optimal dissemination scheme for many cases.

Chapter 4

Optimal Power Allocation for Multiple Transmitters

4.1 Introduction

Power control is one of the most fundamental problems in wireless networks. The rules governing the availability and quality of wireless connections can be described by *physical* or *fading channel* models (cf. [50, 7, 52]). Among those, a commonly studied model is the *signal-to-interference ratio (SIR)* model. [1] In the SIR model, the energy of a signal fades with the distance to the power of the *path-loss parameter* α. Formally, let $\mathrm{d}(r_i, t_j)$ be the Euclidean distance between the receiver r_i and the transmitter t_j, and assume that each transmitter t_i transmits with power X_i. At the location of receiver r_i, the transmission of station t_i is correctly received if

$$\frac{X_i \cdot \mathrm{d}(r_i, t_i)^{-\alpha}}{\sum_{j \neq i} X_j \cdot \mathrm{d}(r_i, t_j)^{-\alpha}} \geq \beta. \tag{4.1}$$

In the basic setting, known as the SISO (Single Input, Single Output) model, we are given a network of n receivers $\{r_1 \ldots r_n\}$ and n transmitters $\{t_1 \ldots t_n\}$ embedded in $\mathbb{R}^d$, where each transmitter is assigned to a single receiver (see Figure 4.1 (a) for an

[1]This is a special case of the *signal-to-interference & noise ratio (SINR)* model where the noise is zero.

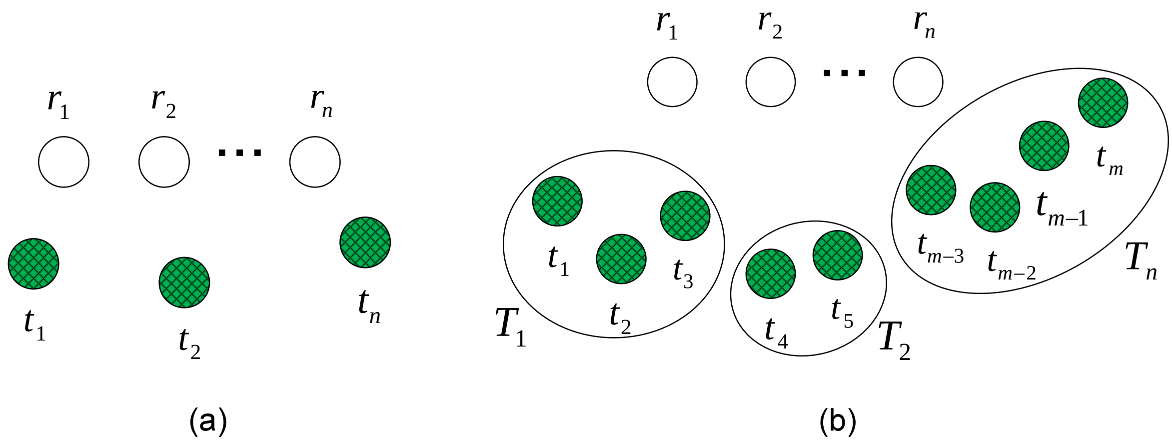


Figure 4.1: (a) – Square system, i.e., each receiver has a single dedicated transmitter (SISO case). (b) – Nonsquare system, i.e., each receiver has multiple dedicated transmitters (MISO case).

illustration). The main question then is to find the optimal (i.e., largest) $\beta^* = \max_{\overline{X}} \beta$ and the power assignment $\overline{X}^*$ that achieves it when we consider Eq. (4.1) at each receiver r_i. The larger β is, the simpler (and cheaper) is the hardware implementation required to decode messages in a wireless device.

We now turn to our main problem – optimal power allocation in MISO (Multiple Input Single Output) systems. In the MISO setting, a set of multiple synchronized transmitters, located at different places, can transmit at the same time to the same receiver (see Figure 4.1 (b) for an illustration). Formally, for each receiver r_i we have a set of k_i transmitters, of a total of m transmitters. Let T represent the set of all m transmitters, and T_i represent the set of k_i transmitters dedicated to receiver r_i. The signal to interference (SIR) equation at receiver r_i is then:

$$\frac{\sum_{t_j \in T_i} X_j \cdot \mathrm{d}(r_i, t_j)^{-\alpha}}{\sum_{t_j \in T \setminus T_i} X_j \cdot \mathrm{d}(r_i, t_j)^{-\alpha}} \ \geq \ \beta \ . \tag{4.2}$$

In such a setting, we would like to find the best way to organize the power allocation between the transmitters of each receiver. We assume that the signals from the transmitters are perfectly synchronized, so they just sum up at the receiver.

We will show that there exists an optimal power allocation in which only one transmitter per receiver can transmit in order to achieve β^*. In other words, somewhat surprisingly, the option of cooperation (i.e., splitting the available power between multiple transmitters) does not improve the situation, in the sense that in the optimum solution, no cooperation is needed and only one transmitter per receiver needs to work. Hence, the additional power of having several potential transmitters per receiver translates into choosing the "best" single transmitter and not into sharing the available power between the transmitters in some way, as one might have expected.

4.1.1 Related Work of MISO Power Control

In this subsection we highlight the differences between our proposed MISO power-control algorithm and the existing approaches to this problem. The vast literature on power control in MISO and MIMO systems considers mostly the joint optimization of power control with beamforming (which is represented by a precoding and shaping matrix). In the commonly studied *downlink scenario*, a single transmitter with m antennae sends independent information signals to n decentralized receivers. With this formulation, the goal is to find an optimal power vector of length n and an $n \times m$ beamforming matrix. The standard heuristic applied to this problem is an iterative strategy that alternately repeats a *beamforming* step (i.e., optimizing the beamforming matrix while fixing the powers) and a *power control* step (i.e., optimizing powers while fixing the beamforming matrix) until convergence [15, 14, 21, 53, 18]. In [15], the geometric convergence of such a scheme has been established. In addition, [55] formalizes the problem as a conic optimization program that can be solved numerically. In summary, the current algorithms for MIMO power-control (with beamforming) are of numeric and iterative flavor, though with good convergence guarantees. In contrast, the current work considers the simplest MISO setting (without coding techniques) and aims at *characterizing* the mathematical *structure* of the optimum solution. In particular, we establish the fact that the optimal max-min SIR value is an algebraic number (i.e., the root of a characteristic polynomial) and the optimum power vector is a $\mathbf{0^*}$ solution. Equipped with this structure, we design an efficient algorithm that is more accurate than off-the-shelf numeric optimization packages that were usually applied in this context. Needless to say, the structural properties of the optimum solution are of theoretical interest in addition to their applicability.

4.1.2 Overview of Results of the Current Chapter

In this chapter we make two contributions. First, we show how to use the Generalized Perron-Frobenius Theorem presented in [4] to state that for the MISO power allocation problem (where each receiver has multiple dedicated transmitters) there exists an optimal solution in which only one transmitter per receiver transmits (we call such a solution a $\mathbf{0^*}$ solution).

While in the SISO case (single transmitter per each receiver), the optimal solution can be computed in polynomial time (by just finding the Eigen system of an appropriate matrix); this is not clear in the extended (MISO) case, since the corresponding optimization problem (Problem 4.5) is not convex [11] (and also not log-convex as we showed in [4]). Even if we know that the optimal solution can be a $\mathbf{0^*}$ solution (only one transmitter per receiver transmits), there are exponentially many choices even if each receiver has only two transmitters to choose from. So, our second (and the main) contribution here is a polynomial time algorithm to find the optimal SIR β^* and the corresponding power allocation $\overline{X}^*$. The algorithm uses the fact that for a given β we get a relaxed problem that is convex (actually it becomes linear). This allows us to employ the well-known ellipsoid method [42] for testing a specific β for feasibility. Hence, the problem reduces to finding the maximum feasible β, and the algorithm does so by applying binary search on β. Clearly, the search results in an approximate solution. Obtaining an exact optimal β^*, along with an appropriate vector $\overline{X}^*$, is another challenging aspect of the algorithm, which is successfully solved via an original approach based on the extended PF Theorem [4]. Finally, we prove that the proposed algorithm is polynomial.

4.2 Optimal Power Allocation using Perron-Frobenius Theorem

4.2.1 Preliminaries

Consider a set of n receivers $R = \{r_1, \ldots, r_n\}$ and a set of m $(m \geq n)$ transmitters $T = \{t_1, \ldots, t_n\}$. We say that transmitter t_j is dedicated to receiver r_i if the signal of t_j is the desired signal for r_i and not an interference. Each receiver r_i $(i \in [1, \ldots, n])$ has k_i $(k_i \geq 1)$ dedicated transmitters, so $\sum_{i=1}^{n} k_i = m$. We denote the set of all transmitters dedicated to the receiver r_i as T_i; consequently, $|T_i| = k_i$. We assume that all the transmitters dedicated to the same receiver transmit the same information and are perfectly synchronized; therefore, their powers (multiplied by the appropriate gains) sums up at the receiver.

Let $\mathrm{d}(r_i, t_j)$ be the Euclidean distance between the receiver r_i and the transmitter t_j. We assume that the signal transmitted by t_j is received by r_i with the gain $\mathrm{d}(r_i, t_j)^{-\alpha}$, where α is the path-loss parameter and usually equals 2. We denote gain by $g(i, j)$, which means that $g(i, j) = \mathrm{d}(r_i, t_j)^{-\alpha}$. Now we can define a system $\mathcal{L} = \langle A, B \rangle$ as a pair of matrices $A, B \in \mathbb{R}^{n \times m}$ that captures the mutual gains between all the receiver-transmitter pairs, and the information regarding which transmitter is dedicated to which receiver. Following are the definitions of the matrices:

$$A(i,j) = \begin{cases} g(i,j) & \text{if } t_j \notin T_i \\ 0 & \text{if } t_j \in T_i \end{cases} \tag{4.3}$$

$$B(i,j) = \begin{cases} g(i,j) & \text{if } t_j \in T_i \\ 0 & \text{if } t_j \notin T_i \end{cases} \tag{4.4}$$

It is easy to see that row i in A represents all the interference to the receiver r_i, while row i in B represents all the desired signals (i.e., signals from transmitters that are dedicated to r_i). A system $\mathcal{L} = \langle A, B \rangle$ is called a *square system* if $m = n$ (i.e., each receiver has exactly one dedicated transmitter), and such a square system is denoted by $\mathcal{L}^s$. Notice that the matrices A, B in $\mathcal{L}^s$ are square $n \times n$ matrices, and we assume that, in square systems, t_j is dedicated to r_j ($j \in [1 \ldots, n]$); hence, B is diagonal.

One can also obtain a square system $\mathcal{L}^s$ from a non-square system $\mathcal{L}$ by selecting a single dedicated transmitter for each receiver and eliminating all the other transmitters. In this case, $\mathcal{L}^s$ is called a *square sub-system* of $\mathcal{L}$. Notice that there are many possible square sub-systems, since there are many possible options to select a single dedicated transmitter for each receiver. In a square sub-system (as in a square system) we will assume that t_j is dedicated to r_j (i.e., B is diagonal), which can be achieved by simply renaming the transmitters. For the above description, we can understand that the MISO (Multiple Input Single Output) case, mentioned in the Introduction, corresponds to a non-square system, while the SISO (Single Input Single Output) corresponds to a square system.

The following irreducibility definitions were introduced in [4] and are necessary for applying the Perron-Frobenius Theorem. A square system $\mathcal{L}^s = \langle A, B \rangle$ is irreducible if B is a nonsingular matrix and A is an irreducible matrix. A system $\mathcal{L}$ is irreducible if all its possible square sub-systems are irreducible.

Observation 4.1. *In the context of the power allocation problem (which is formally determined by Eq. 4.3 and 4.4), systems $\mathcal{L}^s$ and $\mathcal{L}$ are always irreducible.*

Proof. First we show that $\mathcal{L}^s = \langle A, B \rangle$ is irreducible. The matrix B is diagonal and thus nonsingular. Now we show that matrix A is irreducible. In a square system, every receiver has exactly one dedicated transmitter; therefore in row i of A we will find 0 at position i and non-zero values at the other positions. Consider a directed graph associated with A. Clearly, from any vertex, there is a directed edge to any other vertex (except the vertex itself). Thus, the graph is strongly connected, which implies that the matrix is irreducible. Thus, any square system $\mathcal{L}^s$ is irreducible. The last implies that all the square sub-systems of $\mathcal{L}$ are also irreducible (any square sub-system is a square system by itself) and thus, $\mathcal{L}$ is irreducible. □

Problem Formulation. Given a system $\mathcal{L} = \langle A, B \rangle$,

$$\begin{aligned}&\text{maximize } \beta \text{ subject to:}\\ &A \cdot \overline{X} \leq \frac{1}{\beta} \cdot B \cdot \overline{X}, \quad \left\|\overline{X}\right\|_1 = 1, \quad \overline{X} \geq \overline{0}.\end{aligned} \tag{4.5}$$

Where β is the signal to interference (SIR) ratio that should be satisfied at each receiver. The optimal (maximal) value of β is denoted by β^*, while the maximization is performed over all possible power allocation vectors $\overline{X}$, i.e., $\beta^* = \max_{\overline{X}} \beta$. Power allocation that achieves β^* is denoted by $\overline{X}^*$. Notice also the normalization constraint $\left\|\overline{X}\right\|_1 = 1$, which prevents us from having an infinite number of power allocations for a given SIR β. The SIR constraint $A \cdot \overline{X} \leq \frac{1}{\beta} \cdot B \cdot \overline{X}$ can be understood by looking at the standard SIR equation for each receiver r_i:

$$\frac{\text{desired signal to } r_i}{\text{interference to } r_i} = \frac{\sum_{t_j \in T_i} X_j \cdot g(i,j)}{\sum_{t_j \in T \setminus T_i} X_j \cdot g(i,j)} \geq \beta \ , \tag{4.6}$$

and combining them in a matrix form. If the optimal power vector $\overline{X}^*$ appears to have exactly n non-zero values, such a solution means that for each receiver there is exactly one dedicated transmitter, and we denote such a solution - a $\mathbf{0^*}$ solution.

4.2.2 Optimal Power Allocation for Square Systems

As it turns out, the power control problem for square systems can be solved elegantly using the Perron-Frobenius (PF) Theorem, as was proposed in a seminal work of Zander [56]. Consider the following optimization problem ($Z \in \mathbb{R}^{n \times n}$):

$$\text{maximize } \beta \text{ subject to:} \tag{4.7}$$
$$Z \cdot \overline{X} \leq \frac{1}{\beta} \cdot \overline{X}, \quad \left\|\overline{X}\right\|_1 = 1, \quad \overline{X} \geq \overline{0}.$$

Let β^* denote the optimal solution for Program 4.7. The Perron-Frobenius (PF) Theorem characterizes the solution to this optimization problem and shows the following:

Theorem 4.1 (PF Theorem, short version)**.** *Let Z be an irreducible nonnegative matrix. Then $\beta^* = 1/r$, where $r \in \mathbb{R}_{>0}$ is the largest Eigen value of Z, called the* Perron–Frobenius (PF) root *of Z. There exists a unique (Eigen-)vector $\overline{\mathbf{P}} > \overline{0}$, $\left\|\overline{\mathbf{P}}\right\|_1 = 1$, such that $Z \cdot \overline{\mathbf{P}} = r \cdot \overline{\mathbf{P}}$, called the* Perron vector *of Z.*

So, given a square system $\mathcal{L}^s = \langle A, B \rangle$ we can define a matrix $Z = B^{-1} \cdot A$ (this is possible since B is diagonal in any square system). Clearly, if A is irreducible, so is Z. Thus, by rewriting the Problem 4.5 with Z instead of $(B^{-1} \cdot A)$ we obtain Problem 4.7, which can be simply solved by Theorem 4.1.

4.2.3 Optimal Power Allocation for Nonsquare Systems

In [4] we extended Theorem 4.1 to nonsquare matrices. Consider the following extended optimization problem. (Here $A, B \in \mathbb{R}^{n \times m}$.)

$$\text{maximize } \beta \text{ subject to:} \tag{4.8}$$
$$A \cdot \overline{X} \leq \frac{1}{\beta} \cdot B \cdot \overline{X}, \quad \left\|\overline{X}\right\|_1 = 1, \quad \overline{X} \geq \overline{0}.$$

The problem for square systems (SISO case with a single transmitter per receiver) was defined for a square matrix, so the rise of Eigen values seems natural. In contrast, in the generalized setting, the situation seems more complex. In [4] we showed an extension of the PF Theorem to nonsquare matrices and systems.

Theorem 4.2 (Multiple Choice PF Theorem, short version [4])**.** *Let* $\langle A, B \rangle$ *be an irreducible nonnegative system. Then* $\beta^* = 1/r$*, where* $r \in \mathbb{R}_{>0}$ *is the largest* Perron–Frobenius (PF) root *of all* $n \times n$ *square sub-systems. There exists a vector* $\overline{\mathbf{P}} \geq 0$ *such that* $A \cdot \overline{\mathbf{P}} = r \cdot B \cdot \overline{\mathbf{P}}$ *and* $\overline{\mathbf{P}}$ *has* n *entries greater than 0 and* $m - n$ *entries equaling 0 (referred to as a* $\mathbf{0^*}$ *solution).*

Since the system $\mathcal{L}$ is irreducible (see Observation 4.1), we can apply Theorem 4.2 to conclude that there exists an optimal $\mathbf{0^*}$ solution for our problem. In the next section we will show how to actually find this solution (i.e., the optimal SIR β^* and the optimal power vector $\overline{X}$) in polynomial time.

4.3 Computing the Optimal Solution

In this section we present a polynomial time algorithm (Algorithm Comp-$\mathbf{0^*}$) for computing the optimal $\mathbf{0^*}$ solution for the MISO power allocation problem.

The method By Theorem 4.2, there exists an optimal $\mathbf{0^*}$ solution for Program (4.5) with $\beta = \beta^*$. For ease of analysis, we assume that the gains are integral, i.e., $g(i,j) \in \mathbb{Z}^+$, for every $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$. If this does not hold, then the gains can be rounded or scaled to achieve this. Let

$$\mathcal{G}_{max}(\mathcal{L}) = \max_{i \in \{1,\ldots,n\}, j \in \{1,\ldots,m\}} \{g(i,j)\}, \tag{4.9}$$

and define T_{ellips} as the running time of the Ellipsoid method [42] for the following optimization program, which becomes linear due to the fact that β is no longer a variable:

$$\text{maximize } 1 \text{ subject to:} \quad A \cdot \overline{X} \leq \frac{1}{\beta} \cdot B \cdot \overline{X}, \quad \left\|\overline{X}\right\|_1 = 1, \quad \overline{X} \geq \overline{0}. \tag{4.10}$$

Recall that are concerned with an exact optimal solution for a non-convex optimization problem (see Program (4.5)). Using the convex relaxation of Program (4.10), a binary search can be applied for finding an approximate solution up to a predefined accuracy. The main challenge is then to find (a) an optimal solution (and not an approximate solution), and (b) among all the optimal solutions, to find one that is a $\mathbf{0}^*$ solution. Notice that the number of possible allocations where each receiver has a single active transmitter is exponentially large (even when every receiver has only two dedicated transmitters.)

Here we prove the following.

Theorem 4.3. *The optimal* $\mathbf{0}^*$ *solution for the MISO power allocation problem can be computed in time* $O(n^3 \cdot T_{ellips} \cdot (\log(n \cdot \mathcal{G}_{max}) + n))$.

Let

$$\Delta_\beta \;=\; (n\mathcal{G}_{max})^{-8n^3}. \tag{4.11}$$

The key observation in this context is the following.

Lemma 4.1. *Let* $\mathcal{L}_1^s$ *and* $\mathcal{L}_2^s$ *be any two square sub-systems of a system* $\mathcal{L}$*, with optimal SIR values* β_1 *and* β_2*, respectively. Then (assuming* $\beta_1 \neq \beta_2$*),*

$$|\beta_1 - \beta_2| > \Delta_\beta.$$

By performing a polynomial number of steps of a binary search for the optimal β^*, one can converge to a value β^- that is at most Δ_β far from β^*, i.e., $\beta^* - \beta^- \leq \Delta_\beta$. Let $Range_{\beta^*} = [\beta^-, \beta^*]$. Then by Theorem 4.2, we are guaranteed that for any square sub-system $\mathcal{L}^s$ such that $\beta^*(\mathcal{L}^s) \in Range_{\beta^*}$, it holds that $\beta^*(\mathcal{L}^s) = \beta^*$. To prove Lemma 4.1, we first show that there is a lower bound on the difference between *any* two different PF Eigenvalues of any two square sub-systems, i.e., we show that for any two square sub-systems $\mathcal{L}_1^s, \mathcal{L}_2^s$, their PF roots r_1 and r_2 cannot be too close if they are different.

Recall that for a square sub-system $\mathcal{L}^s = \langle A, B \rangle$ we define a matrix $Z(\mathcal{L}^s) = B^{-1} \cdot A$, where B can be considered to be diagonal (receiver r_i has exactly one dedicated transmitter t_i). We begin the analysis by scaling the entries of $Z(\mathcal{L}^s)$ to obtain an integer-valued matrix Z^{int}. The scaling is needed in order to employ a well-known bound on the minimal distance between the roots of integer polynomials presented in the following lemma. The naïve height of an integer polynomial is the maximum of the absolute values of its coefficients.

Lemma 4.2 (Bugeaud and Mignotte in [13]). *Let $P(X)$ and $Q(X)$ be nonconstant integer polynomials of degree n and m, respectively. Denote by r_P and r_Q a zero of $P(X)$ and $Q(X)$, and $H(P)$ and $H(Q)$ the naïve heights of $P(X)$ and $Q(X)$, respectively. Assuming that $P(r_Q) \neq 0$, we have:*

$$| r_P - r_Q| \;\; \geq \;\; 2^{1-n}(n+1)^{\frac{1}{2}-m}(m+1)^{-\frac{n}{2}} H(P)^{-m} H(Q)^{-n} \; .$$

We now turn to prove Lemma 4.1.

Proof of Lemma 4.1. Recall that $Z(\mathcal{L}^s) = B^{-1} \cdot A$, therefore,

$$Z(\mathcal{L}^s)_{i,j} = \begin{cases} g(i,j)/g(i,i) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases},$$

where $g(i,i)$ correspond to the gain of the unique transmitter t_i of receiver r_i.

Let us denote $Z_1 = Z(\mathcal{L}_1^s)$, $Z_2 = Z(\mathcal{L}_2^s)$, and r_1 and r_2 will be the PF roots of Z_1 and Z_2, respectively. To employ Lemma 4.2, we first scale Z_1 and Z_2 to obtain two integer-valued matrices Z_1^{int} and Z_2^{int}. The new matrix Z_b^{int}, for $b \in \{1, 2\}$, is constructed by multiplying each entry of Z_b by the common denominator of its entries, i.e., $Z_b^{\text{int}}(i,j) = Z_b(i,j) \cdot \prod_i (g(i,i_1) \cdot g(i,i_2))$, where i_b ($b \in \{1,2\}$) is the index of the single transmitter of receiver r_i in Z_b. Thus, all entries of Z_b^{int} are integers and bounded by $\mathcal{G}_{max}^{2n}$ (since $g(i,j) \leq \mathcal{G}_{max}$).

Let $P_1(x) = \det(x \cdot I - Z_1^{\text{int}})$ and $P_2(x) = \det(x \cdot I - Z_2^{\text{int}})$ be the characteristic polynomials of the matrices Z_1^{int} and Z_2^{int}, respectively (I is the $n \times n$ identity matrix). Note that $P_1(x)$ and $P_2(x)$ are integer polynomials of degree n, and $H(P_1), H(P_2) \leq \mathcal{G}_{max}^{2n^2}$ (since $|\det(Z_b^{\text{int}})| \leq (\mathcal{G}_{max}^{2n})^n$). Let r_1^{int} and r_2^{int} correspond to the PF Eigenvalues of Z_1^{int} and Z_2^{int}, respectively. Thanks to Lemma 4.2 we obtain:

$$|r_1^{\text{int}} - r_2^{\text{int}}| \geq 2^{1-n}(n+1)^{\frac{1}{2}-n}(n+1)^{-\frac{n}{2}}(\mathcal{G}_{max}^{2n^2})^{-n}(\mathcal{G}_{max}^{2n^2})^{-n} \ = \ 2^{1-n}(n+1)^{\frac{1-3n}{2}}\mathcal{G}_{max}^{-4n^3}.$$

Finally, by definition of Z_b^{int},

$$|r_1^{\text{int}} - r_2^{\text{int}}| \ = \ |r_1 - r_2| \prod_i (g(i,i_1) \cdot g(i,i_2)),$$

and thus

$$|r_1 - r_2| \ \geq \ \frac{2^{1-n}(n+1)^{\frac{1-3n}{2}}\mathcal{G}_{max}^{-4n^3}}{\prod_i (g(i,i_1) \cdot g(i,i_2))} \ \geq \ \frac{2^{1-n}(n+1)^{\frac{1-3n}{2}}\mathcal{G}_{max}^{-4n^3}}{\mathcal{G}_{max}^{2n}} \ \geq \ (n\mathcal{G}_{max})^{-6n^3}.$$

We now turn to translate the distance between r_1 and r_2 into the distance between $1/r_1$ and $1/r_2$ (corresponding to the optimal SIR values β_1 and β_2 of the square sub-systems $\mathcal{L}_1^s$ and $\mathcal{L}_2^s$, respectively). The next auxiliary claim gives a bound for $\lambda \in EigVal(Z)$ as a function of $\mathcal{G}_{max}$.

Claim 4.1. *Let λ be an Eigenvalue of an $n \times n$ matrix Z such that $|Z(i,j)| \leq \mathcal{G}_{max}$. Then $|r| \leq n\mathcal{G}_{max}$.*

Proof. Let $\overline{X}$ be the Eigenvector of Z and assume that $||\overline{X}||_2 = 1$. Since $\overline{X}^T \cdot Z \cdot \overline{X} =$

$\lambda\overline{X}^T \cdot \overline{X} = \lambda$, we have:

$$
\begin{aligned}
|\lambda| &= |\overline{X}^T Z \overline{X}| \\
&= |\sum_i \sum_j X(i) Z(i,j) X(j)| \\
&\leq \mathcal{G}_{max} \cdot |\sum_i \sum_j X(i) \cdot X(j)| \\
&= \mathcal{G}_{max} \cdot |\sum_i X(i)| \cdot |\sum_j X(j)| \\
&= \mathcal{G}_{max} \cdot \|\overline{X}\|_1^2.
\end{aligned}
$$

Now we show that $\|\overline{X}\|_1^2 \leq (\sqrt{n}\|\overline{X}\|_2)^2$.

$$
\begin{aligned}
(\sqrt{n}\|\overline{X}\|_2)^2 - \|\overline{X}\|_1^2 &= n(x_1^2 + x_2^2 + \cdots + x_n^2) - (x_1 + x_2 + \cdots + x_n)^2 \\
&= (n-1)x_1^2 + (n-1)x_2^2 + \cdots + (n-1)x_n^2 \\
&- 2x_1x_2 - 2x_1x_3 - \cdots - 2x_{n-1}x_n \\
&= (x_1 - x_2)^2 - (x_1 - x_3)^2 - \cdots - (x_{n-1} - x_n)^2 \geq 0.
\end{aligned}
$$

Hence,

$$
\begin{aligned}
|\lambda| &\leq \mathcal{G}_{max} \cdot \|\overline{X}\|_1^2 \\
&\leq \mathcal{G}_{max} \cdot (\sqrt{n}\|\overline{X}\|_2)^2 = n\mathcal{G}_{max}.
\end{aligned}
$$

□

Now, we can finish the proof of Lemma 4.1. Since $r_1, r_2 \in (0, n\mathcal{G}_{max}]$, it follows that:

$$
|\beta_2 - \beta_1| = \left|\frac{1}{r_2} - \frac{1}{r_1}\right| = \left|\frac{r_1 - r_2}{r_1 r_2}\right| \geq \frac{|r_1 - r_2|}{(n\mathcal{G}_{max})^2} \geq (n\mathcal{G}_{max})^{-8n^3}.
$$

□

4.3.1 Algorithm description

We now describe the algorithm for computation of the optimal **0*** solution. The algorithm comprises two phases. In the first phase, the algorithm searches for the nearly optimal feasible SIR, namely for the β^-, such that $\beta^* - \beta^- \leq \Delta_\beta$. Then, in the second phase, the algorithm iteratively constructs the square sub-system $\mathcal{L}^s$ that achieves β^-, which means that $\mathcal{L}^s$ also achieves β^*. The last is true since, due to Theorem 4.2, there exists a square sub-system that achieves β^*, and from Lemma 4.1 we know that any square system that achieves SIR that is less than Δ_β close to β^* must also achieve β^*.

We define a function that will indicate whether a given SIR β is feasible for the given system (i.e., for the optimization Problem 4.10):

$$f(\beta, \mathcal{L}) = \begin{cases} 1, & \text{if there exists } \overline{X} \text{ such that } \left\|\overline{X}\right\|_1 = 1, \overline{X} \geq \overline{0}, \text{ and} \\ & A \cdot \overline{X} \leq 1/\beta \cdot B \cdot \overline{X} \ , \\ 0, & \text{otherwise.} \end{cases}$$

Note that f can be computed in polynomial time using the Ellipsoid method.

The pseudocode of the algorithm is presented formally in Figure 4.2.

To establish Theorem 4.3, we prove the correctness of Algorithm Comp-**0*** and bound its runtime. We begin with two auxiliary claims.

Claim 4.2. $\beta^*(\mathcal{L}) \leq \mathcal{G}_{max}$.

Proof. Consider a square sub-system of $\mathcal{L}$ that achieves β^*. Now, assuming $n > 2$, we remove all the transmitter-receiver pairs except the two: (r_1, t_1) and (r_2, t_2). Clearly, optimal SIR β' for this 2-pairs square system can be only larger than β^*. So, we obtain:

$$\beta^* \leq \beta' = \min\left\{\frac{g(1,1) \cdot X_1}{g(1,2) \cdot X_2}, \frac{g(2,2) \cdot X_2}{g(2,1) \cdot X_1}\right\} \leq \min\left\{\frac{\mathcal{G}_{max} \cdot X_1}{X_2}, \frac{\mathcal{G}_{max} \cdot X_2}{X_1}\right\}.$$

Algorithm Comp-$\mathbf{0}^*$

/* Binary search phase: finding β^- such that $\beta^* - \beta^- < \Delta_\beta$ */

1. $\beta \leftarrow 1$;
2. While $f(\beta, \mathcal{L}) = 1$ do:
   $\beta \leftarrow 2\beta$;
3. If $\beta > 1$, then $\beta^- \leftarrow \beta/2$, else $\beta^- \leftarrow 0$;
4. $\beta^+ \leftarrow \beta$;
5. While $\beta^+ - \beta^- \geq \Delta_\beta$ do: /* from now on $\beta^- \leq \beta^* < \beta^+$ */
   (a) $\beta \leftarrow (\beta^- + \beta^+)/2$;
   (b) If $f(\beta, \mathcal{L}) = 1$, then $\beta^- \leftarrow \beta$, else $\beta^+ \leftarrow \beta$;

/* Transmitters elimination phase: finding a $\mathbf{0}^*$ solution */

6. $\mathcal{L}_1 \leftarrow \mathcal{L}$;
7. For $i = 1$ to n do:
   (a) Remove from $\mathcal{L}_i$ all the transmitters dedicated to the receiver r_i except one, such that: $f(\beta^-, \mathcal{L}_i) = 1$;
   (b) $\mathcal{L}_{i+1} \leftarrow \mathcal{L}_i$;

/* $\mathcal{L}_n$ is a square sub-system of $\mathcal{L}$ */

8. $\beta^* \leftarrow 1/r$; $\overline{X}^* \leftarrow \overline{\mathbf{P}}$;

Figure 4.2: Pseudocode of Algorithm Comp-$\mathbf{0}^*$.

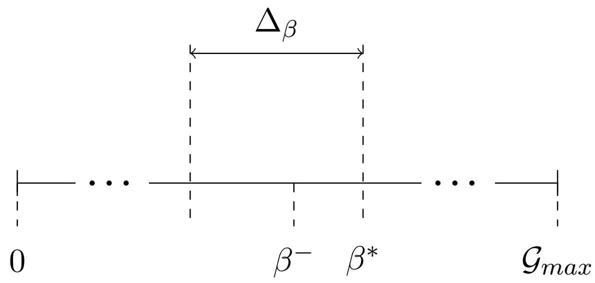


Figure 4.3: By the end of the first phase, the algorithm finds β^-.From Lemma 4.1 and Theorem 4.2, we can conclude that any square system that satisfies β, which is at most Δ_β lower than β^*, satisfies also β^* (i.e., such a square system corresponds to an optimal $\mathbf{0^*}$ solution).

Clearly, $\min\left\{\frac{X_1}{X_2}, \frac{X_2}{X_1}\right\} \leq 1$, thus we get $\beta^*(\mathcal{L}) \leq \mathcal{G}_{max}$ as required. □

Claim 4.3. *By the end of phase 1, Alg. Comp-*$\mathbf{0^*}$ *finds* β^- *such that* $\beta^*(\mathcal{L}) - \beta^- \leq \Delta_\beta$.

Proof. Clearly, $f(\beta, \mathcal{L}) = 1$ for every $\beta \in (0, \beta^*]$. By steps 3 and 5b in Alg. Comp-$\mathbf{0^*}$ (Figure 4.2), we have that $f(\beta^-, \mathcal{L}) = 1$. Therefore $\beta^- < \beta^*(\mathcal{L})$. Note that by the stopping criterion of step 5, we are in a situation where $f(\beta^+, \mathcal{L}) = 0$, $f(\beta^-, \mathcal{L}) = 1$ and $\beta^+ - \beta^- \leq \Delta_\beta$. This implies that $\beta^* \in [\beta^-, \beta^+)$ as required. The claim follows. □

We are now ready to complete the proof of Theorem 4.3.

Proof of Theorem 4.3. We show that Alg. Comp-$\mathbf{0^*}$ satisfies the requirements of the theorem. Note that at the beginning of phase 2 of Alg. Comp-$\mathbf{0^*}$, the computed value β^- is at most Δ_β apart from β^* (see Figure 4.3 for an illustration). In phase 2, the algorithm iteratively constructs a square sub-system of $\mathcal{L}$. At iteration i of the for-loop, we start with a system $\mathcal{L}_i$, for which β^- is feasible, and eliminate all the transmitters of receiver r_i except one. The single transmitter that will remain has to satisfy system feasibility for β^-. Assume by contradiction, that there is no single transmitter satisfying the feasibility condition. Then, it means that there is no $\mathbf{0^*}$ solution achieving β^- for the system $\mathcal{L}_i$, which contradicts Theorem 4.2. Thus, at every iteration $i \in [1, \ldots, n]$ the algorithm will find a single transmitter for receiver r_i that will satisfy system feasibility for β^-. (See Figure 4.4 for an illustration of the algorithm's step 7(a).)

By the end of the algorithm, we get a square sub-system $\mathcal{L}_n$ that is feasible for β^-, and thus, according to Lemma 4.1, achieves β^*. The optimal power allocation $\overline{X}$ is found using Theorem 4.1. Notice that before using Theorem 4.1, we rename the transmitters in the system $\mathcal{L}_n$ so that t_j is dedicated to r_j ($j \in [1, \ldots, n]$) and thus matrix B becomes diagonal, allowing us to obtain the required matrix $Z = B^{-1} \cdot A$ (Figure 4.5 shows the renamed system). Hence, the correctness of the algorithm is established.

Finally, we analyze the runtime of the algorithm. Note that there are

$$O(\log{(\beta^*(\mathcal{L})/\Delta_\beta)} + n)$$

calls for the Ellipsoid method (computing $f(\beta^-, \mathcal{L}_i)$), namely, $O(\log{(\beta^*(\mathcal{L})/\Delta_\beta)})$ calls in the first phase and n calls in the second phase. By plugging Eq. (4.9) in Claim 4.2, Theorem 4.3 follows. □

4.4 Conclusions

In this chapter we gave a solution to the power allocation problem with multiple transmitters (also known as the MISO system). First, using the Generalized (Multiple Choice) Perron-Frobenius Theorem, proved in [4], we can state that there exists an optimal solution in which only one transmitter for each receiver can transmit (we call such a solution – a $\mathbf{0}^*$ solution). Then, we gave a polynomial time algorithm for finding the optimal SIR value β^* and the corresponding power allocation vector $\overline{X}^*$. We proved the correctness and the running time of the algorithm.

We note that our result, regarding the existence of the optimal $\mathbf{0}^*$ solution, is (somewhat) in contradiction to the well-established fact that MISO and MIMO

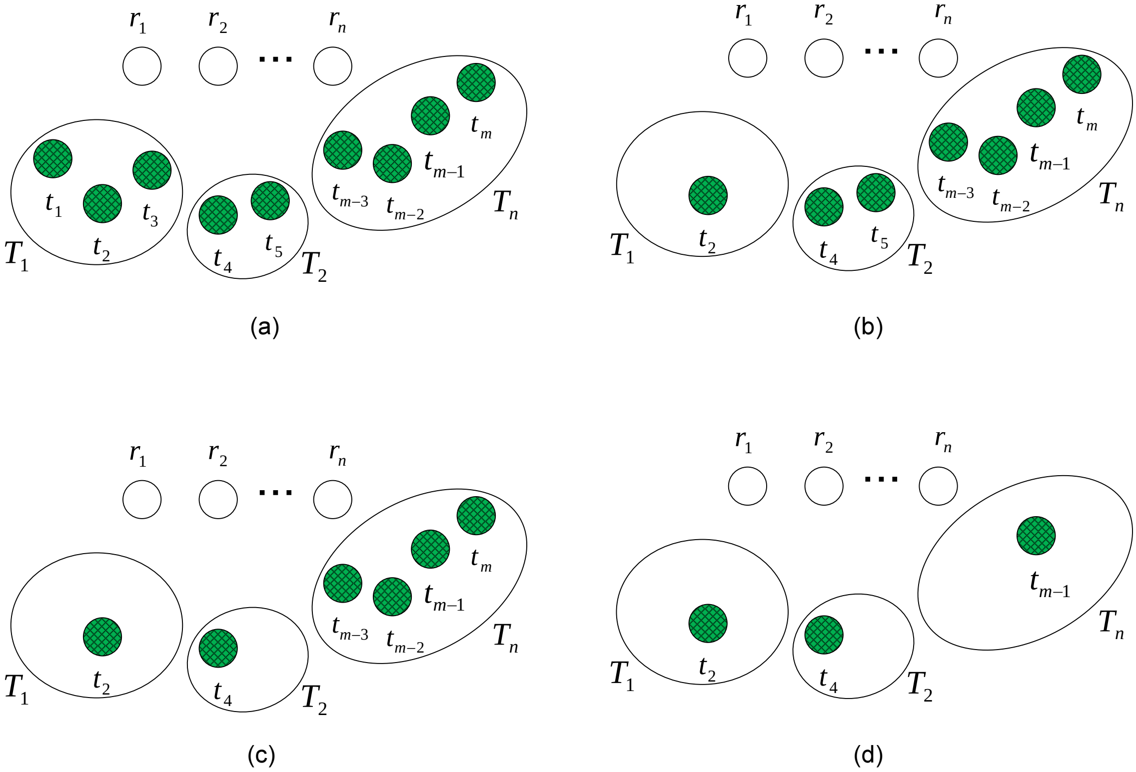


Figure 4.4: (a) Example of system $\mathcal{L}_1$ (initial given system). (b) Example of system $\mathcal{L}_2$. Receiver r_1 is left with a single active transmitter. (c) Example of system $\mathcal{L}_3$. Receivers r_1 and r_2 each has a single active transmitter. (d) Example of system $\mathcal{L}_n$, which is a square system since each receiver has exactly one dedicated active transmitter.

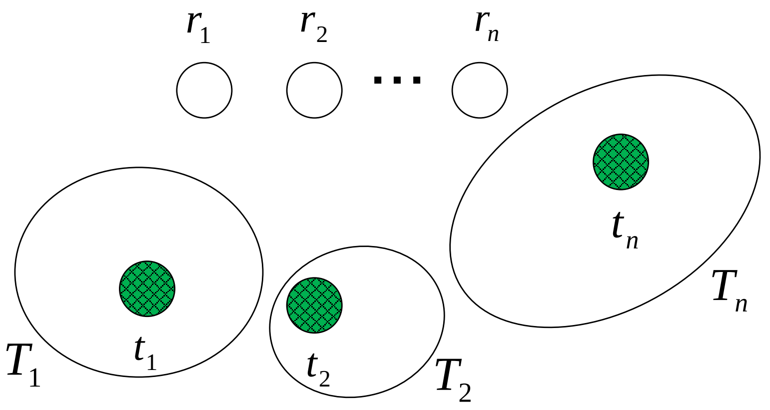


Figure 4.5: Renaming of system $\mathcal{L}_n$ so that t_j is dedicated to r_j ($j \in [1, \dots, n]$).

(Multiple Input Multiple Output) systems, where transmitters transmit in parallel, do improve the capacity of wireless networks, which corresponds to increasing β^* [28]. There are several reasons for this apparent dichotomy, but they are all related to the simplicity of our SIR model. For example, if the ratio between the maximal power to the minimum power is bounded, then our result does not hold any more (as proved in [4]). In addition, our model does not capture random noise and small scale fading and scattering [28], which are essential for the benefits of a MIMO system to manifest themselves.

References

[1] S. Angelopoulos, B. Doerr, A. Huber, and K. Panagiotou. Tight bounds for quasirandom rumor spreading. *The Electronic Journal of Combinatorics*, 16(1):R102,1–R102,19, 2009.

[2] C. Avin, M. Borokhovich, K. Censor-Hillel, and Z. Lotker. Order optimal information spreading using algebraic gossip. In *Proceedings of the 30th annual ACM SIGACT-SIGOPS symposium on Principles of distributed computing*, PODC '11, pages 363–372, New York, NY, USA, 2011. ACM.

[3] C. Avin, M. Borokhovich, K. Censor-Hillel, and Z. Lotker. Order optimal information spreading using algebraic gossip. *Distributed Computing*, 26(2):99–117, 2013.

[4] C. Avin, M. Borokhovich, Y. Haddad, E. Kantor, Z. Lotker, M. Parter, and D. Peleg. Generalized perron-frobenius theorem for multiple choice matrices, and applications. In *SODA*, pages 478–497, 2013.

[5] C. Avin, A. Cohen, Y. Haddad, E. Kantor, Z. Lotker, M. Parter, and D. Peleg. Sinr diagram with interference cancellation. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '12, pages 502–515. SIAM, 2012.

[6] C. Avin, Y. Emek, E. Kantor, Z. Lotker, D. Peleg, and L. Roditty. Sinr diagrams: Convexity and its applications in wireless networks. *J. ACM*, 59(4):18:1–18:34, Aug. 2012.

[7] U. Black. *Mobile and Wireless Networks.* Prentice Hall, 1996.

[8] M. Borokhovich, C. Avin, and Z. Lotker. Tight bounds for algebraic gossip on graphs. In *ISIT*, pages 1758 –1762, 2010.

[9] M. Borokhovich, C. Avin, and Z. Lotker. Bounds for algebraic gossip on graphs. *Random Structures & Algorithms*, page doi:10.1002/rsa.20480, 2012.

[10] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, June 2006.

[11] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, New York, NY, USA, 2004.

[12] S. P. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Gossip algorithms: design, analysis and applications. In *INFOCOM*, pages 1653–1664, 2005.

[13] Y. Bugeaud and M. Mignotte. On the distance between roots of integer polynomials. *Proceedings of the Edinburgh Mathematical Society (Series 2)*, 47:553–556, 10 2004.

[14] D. Cai, T. Quek, and C. W. Tan. A unified analysis of max-min weighted sinr for mimo downlink system. *Signal Processing, IEEE Transactions on*, 59(8):3850 –3862, aug. 2011.

[15] D. W. H. Cai, T. Q. S. Quek, and C. W. Tan. A unified analysis of max-min weighted SINR for MIMO downlink system. *IEEE Trans. Signal Processing*, 59(8), 2011.

[16] K. Censor-Hillel, B. Haeupler, J. Kelner, and P. Maymounkov. Global computation in a poorly connected world: fast rumor spreading with no dependence on conductance. In *Proceedings of the 44th symposium on Theory of Computing*, STOC '12, pages 961–970, New York, NY, USA, 2012. ACM.

[17] A. Chaintreau, P. Fraigniaud, and E. Lebhar. Opportunistic spatial gossip over mobile social networks. In *WOSP '08: Proceedings of the first workshop on Online social networks*, pages 73–78, New York, NY, USA, 2008. ACM.

[18] M. C. Chee Wei Tan and R. Srikant. Maximizing sum rate and minimizing mse on multiuser downlink: Optimality, fast algorithms, and equivalence via max-min sinr. *IEEE Transactions on Signal Processing*, 59(12):6127–6143, 2011.

[19] H. Chen and D. Yao. *Fundamentals of Queueing Networks: Performance, Asymptotics, and Optimization*, volume 46 of *Applications of Mathematics*. Springer-Verlag, New York, first edition, 2001.

[20] H. Chen and D. Yao. *Fundamentals of Queueing Networks: Performance, Asymptotics, and Optimization*, volume 46 of *Applications of Mathematics*. Springer-Verlag, New York, first edition, 2001.

[21] M. Chiang, P. Hande, T. Lan, and C.-W. Tan. Power control in wireless cellular networks. *Foundations and Trends in Networking*, 2(4):381–533, 2008.

[22] S. Deb, M. Médard, and C. Choute. Algebraic gossip: a network coding approach to optimal multiple rumor mongering. *IEEE Transactions on Information Theory*, 52(6):2486–2507, 2006.

[23] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels. Dynamo: amazon's highly available key-value store. *SIGOPS Oper. Syst. Rev.*, 41(6):205–220, Oct. 2007.

[24] A. J. Demers, D. H. Greene, C. Hauser, W. Irish, J. Larson, S. Shenker, H. E. Sturgis, D. C. Swinehart, and D. B. Terry. Epidemic algorithms for replicated database maintenance. *Operating Systems Review*, 22(1):8–32, 1988.

[25] B. Doerr, T. Friedrich, and T. Sauerwald. Quasirandom rumor spreading. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, SODA '08, pages 773–781, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[26] U. Feige, D. Peleg, P. Raghavan, and E. Upfal. Randomized broadcast in networks. *Random Struct. Algorithms*, 1(4):447–460, 1990.

[27] C. Fernandess and D. Malkhi. On collaborative content distribution using multi-message gossip. In *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*, pages 9 pp.–, 2006.

[28] G. J. Foschini and M. J. Gans. On limits of wireless communications in a fading environment when using multiple antennas. *Wireless Personal Communications*, 6:311–335, 1998.

[29] C. Fragouli, J.-Y. L. Boudec, and J. Widmer. Network coding: an instant primer. *Computer Communication Review*, 36(1):63–68, 2006.

[30] G. F. Frobenius. Uber matrizen aus nicht negativen elementen. *Preussiche Akademie der Wissenschaften zu Berlin*, pages 456–477, 1912.

[31] C. Georgiou, S. Gilbert, R. Guerraoui, and D. R. Kowalski. On the complexity of asynchronous gossip. In *PODC '08: Proceedings of the twenty-seventh ACM symposium on Principles of distributed computing*, pages 135–144, New York, NY, USA, 2008. ACM.

[32] G. R. Grimmett and D. R. Stirzaker. *Probability and random processes*. Oxford University Press, New York, third edition, 2001.

[33] B. Haeupler. Analyzing network coding gossip made easy. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pages 293–302, New York, NY, USA, 2011. ACM.

[34] B. Haeupler. Tighter worst-case bounds on algebraic gossip. *Communications Letters, IEEE*, 16(8):1274 –1276, august 2012.

[35] M. Harchol-Balter, T. Leighton, and D. Lewin. Resource discovery in distributed networks. In *Proceedings of the eighteenth annual ACM symposium on Principles of distributed computing*, PODC '99, pages 229–237, New York, NY, USA, 1999. ACM.

[36] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros. The benefits of coding over routing in a randomized setting. In *ISIT*, pages 442+, 2003.

[37] R. V. D. Hofstad. Random graphs and complex networks. *Lecture Notes*, 2010.

[38] E. Kantor, Z. Lotker, M. Parter, and D. Peleg. The topology of wireless communication. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pages 383–392, New York, NY, USA, 2011. ACM.

[39] R. M. Karp, C. Schindelhauer, S. Shenker, and B. Vöcking. Randomized rumor spreading. In *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 565–574, 2000.

[40] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. In *FOCS*, pages 482–491, 2003.

[41] D. Kempe, J. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146, New York, NY, USA, 2003. ACM.

[42] L. G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademii Nauk SSSR*, 244:1093–1096, 1979.

[43] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. *IEEE Transactions on Information Theory*, 49(2):371–381, 2003.

[44] D. S. Lun, M. Médard, R. Koetter, and M. Effros. On coding for reliable communication over packet networks. *Physical Communication*, 1(1):3 – 20, 2008.

[45] M. Médard and R. Koetter. Beyond routing: An algebraic approach to network coding. In *INFOCOM*, pages 122–130, 2002.

[46] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, New York, NY, USA, 2005.

[47] D. Mosk-Aoyama and D. Shah. Information dissemination via network coding. In *ISIT*, pages 1748–1752, 2006.

[48] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[49] C.-H. Ng and S. Boon-Hee. *Queueing Modelling Fundamentals: With Applications in Communication Networks*. Wiley Publishing, 2008.

[50] K. Pahlavan and A. Levesque. *Wireless information networks*. Wiley, 1995.

[51] O. Perron. Grundlagen fur eine theorie des jacobischen kettenbruchalgorithmus. *Mathematische Annalen*, page 1176, 1907.

[52] T. S. Rappaport. *Wireless Communications-Principles and Practice*. Prentice Hall, 1996.

[53] M. Schubert and H. Boche. Solution of the multiuser downlink beamforming problem with individual SINR constraints. In *IEEE Trans. Vehicular Techno.*, 2004.

[54] D. Shah. Gossip algorithms. *Foundations and Trends in Networking*, 3(1):1–125, 2009.

[55] A. Wiesel, Y. C. Eldar, and S. Shamai. Linear precoding via conic optimization for fixed mimo receivers. *IEEE Transactions on Signal Processing*, 54(1):161–176, 2006.

[56] J. Zander. Performance of optimum transmitter power control in cellular radiosystems. *IEEE Tr. Vehic. Technol.*, 41:57–62, 1992.